

Министерство образования и науки Российской Федерации  
ФГБОУ ВО «Уральский государственный педагогический университет»  
Институт математики, физики, информатики и технологий  
Кафедра физики и математического моделирования

**Проектирование и создание информационной системы  
для торговой организации**  
Выпускная квалификационная работа

Квалификационная работа  
допущена к защите  
Зав.кафедрой ФиММ  
д.ф-м.н., профессор  
Сидоров Валерий Евгеньевич

\_\_\_\_\_  
дата

\_\_\_\_\_  
подпись

Исполнитель:  
Терехов Григорий Юрьевич  
обучающийся БЭ-51z группы

\_\_\_\_\_  
подпись

Руководитель:  
Кошечева Елена Сергеевна,  
к.п.н., доцент кафедры ФиММ

\_\_\_\_\_  
подпись

Екатеринбург 2017

## Содержание

Введение .....	3
Глава 1. Информационные технологии создания систем.....	5
1.1. Особенности информатизации торговой организации.....	5
1.2. Стандарты разработки веб-приложений .....	17
Глава 2. Проектирование и создание информационной системы для торговой организации .....	24
2.1. Разработка серверной части информационной системы.....	24
2.2. Разработка клиентской части информационной системы.....	43
2.3. Тестирование и обзор информационной системы .....	50
Заключение.....	69
Список источников и литературы.....	71
Приложение 1 .....	74

## Введение

В современном мире люди повсеместно используют интернет в качестве универсального и функционального инструмента во многих сферах жизнедеятельности человека. Расширяясь и развиваясь, глобальная сеть получила большую значимость для коммерческой деятельности. Интернет привлёк широкий интерес со стороны бизнеса, благодаря активному формированию электронной коммерции, появлению виртуальных торговых площадок, широким возможностям рекламы и повышению уровня экономического взаимодействия в информационном поле.

Успешность коммерческих организаций всё больше зависит от эффективности и полноты использования информационных и интернет технологий, наиболее актуальным являются информационные системы для накопления, хранения и обработки информации. Такая потребность подтверждается в высоком спросе на рынке соответствующего программного обеспечения. Однако из-за постоянно растущих и меняющихся требований в информатизации и автоматизации, существующие предложения в ряде случаев не могут удовлетворить всех потребностей.

Из примеров можно выделить следующие проблемы, с которыми сталкиваются торговые организации при выборе ПО:

*высокая цена* - гиганты IT индустрии часто предлагают большие и удовлетворяющие текущим потребностям программные комплексы, однако их приобретение и обслуживание, могут позволить организации только с высоким финансированием информатизации;

*нехватка специалистов на рынке труда* – проблема возникает с использованием ИС созданной на базе специфичного или неактуального программного и/или аппаратного обеспечения. Стоимость услуг таких специалистов соответственно может нести значительные затраты;

*низкое качество безопасности* - при критичности и необходимости высокого уровня безопасности выбор на рынке программного обеспечения

становится ещё более меньшим. Небольшие компании-разработчики часто не имеют возможность обеспечить безопасность своей системы на высочайшем уровне, а IT-гиганты с популярными и огромными программными комплексами постоянно сталкиваются со взломом их систем и по мере нахождения слабых мест выпускают обновления безопасности, которые не редко приходят постфактум - после взлома;

*сложность в обслуживании* - может выражаться в необходимости обучения сотрудников для работы в системе, дополнительных и не редко, высоких затратах для поддержания работы, необходимостью в найме дорогих специалистов;

*отсутствие интеграции и современного функционала* - с такой проблемой может столкнуться как малая организация-разработчик из-за недостатка ресурсов, так и крупные организации из-за трудности внедрения новых технологий в большие программные комплексы.

Постоянно развивающиеся интернет-технологии создают новые потребности на рынке программных продуктов и являются главным фактором актуальности разработки новых информационных систем для торговых организаций.

**Цель работы:** проектирование и создание информационной системы для хранения, наполнения и обработки информации для повышения эффективности бизнес-процессов торговой организации.

**Задачи работы:**

1. Составление и анализ актуальных требований информатизации и автоматизации коммерческих организаций.
2. Изучение средств, стандартов и методологий разработки создания информационной системы, решающей комплекс задач в информатизации и автоматизации торговой деятельности.
3. Создание информационной системы для хранения, наполнения и обработки информации, сопутствующей бизнес-процессам.

## **Глава 1. Информационные технологии создания систем**

### **1.1. Особенности информатизации торговой организации**

В системе социальных и экономических связей происходят заметные изменения, вызванные ростом использования интернет-технологий. Новые методы и средства для взаимодействия между людьми и организациями, формируют современную среду ведения бизнес-деятельности, в которой торговым организациям необходимо взаимодействовать с информационными ресурсами, организовывать обмен информацией в реальном времени, мгновенно и независимо от расстояния. Исходя из этого есть необходимость во внедрении новейших способов управления процессами и информацией в среде электронной коммерции, которые будут отвечать современным требованиям рынка [15].

Большая доля прикладных процессов торговой организации составляет работа с реестрами данных [3], таких как: клиенты, поставщики, заказы, товары, а также координация взаимодействия с контрагентами, распределение задач для сотрудников, анализ и аналитика цен, роста и падения продаж, анализ действий по привлечению новых клиентов.

В процессе информирования клиентов важно иметь реестр актуальных данных о товарах, информацию о клиенте, историю его покупок, возможность добавить нового клиента в базу данных, быстро сформировать новый заказ или обработать уже существующий, иметь электронное средство коммуникации [12]. В процессе оформления покупки есть необходимость формирования таких документов, как накладная, торговый чек.

Процессы руководства организации требуют возможность аналитики и анализа поступления прибыли (заказов). Используя базы данных руководитель отдела продаж может проводить аналитику успешности торговли за определённый период времени, прогноз продаж на будущее и анализ эффективности предпринятых ранее действий, к примеру, показатель

успешности новой рекламной кампании. Для этого в определённый момент времени руководитель ОП собирает данные по заказам и контрагентам, анализирует данные и формирует выводы в отчёт деятельности организации, чтобы затем предоставить его своему руководителю — директору. Также руководитель отдела продаж отвечает за работу подчинённых — менеджеров/продавцов, анализируя кол-во успешных и не успешных заказов и иных значимых для бизнеса действий, проделанных работниками в рамках рабочих обязанностей. Директор в свою очередь на основе отчётов проводит аналитику бизнес процессов, например, реализация тех или иных товаров и соответственно необходимость найти большее число поставщиков для этой продукции или необходимость увеличить эту продукцию на складе [12].

Появление такой системы в организации может значительно улучшить показатели по ведению бизнес-процессов. Некоторые организации принимают решение о разработке собственной информационной системы «под ключ», взамен приобретению существующих специализированных программных комплексов, либо использовать альтернативные варианты. Так в качестве альтернативной базы данных в малых организациях выступают привычные электронные таблицы из офисных пакетов. Однако, такой подход вызывает проблемы с составлением отчётов, поиском информации (распределённой в нескольких несвязанных таблицах и даже компьютерах), потерю данных. Такая работа не может быть эффективной и отражать реальные данные, т. к. в процессе сбора, информация может дублироваться, теряться, становиться неактуальной, а вся работа займёт значительное количество времени. Разработка собственной информационной системы может стать верным решением, если при этом проектирование и разработка будут выполняться с использованием современных стандартов и методологий. Такое требование необходимо не только для разработки ИС в качестве конечного продукта, отвечающего текущим требованиям, но и продукта, который сможет развиваться и расширяться соответственно росту организации и рынка. Это также значит, что в будущем над данной системой

будут работать и другие специалисты, эффективность и взаимодействие которых во многом будет зависеть от изначально разработанной основы ИС. Другой немаловажный фактор при создании информационной системы – это выбор обеспечивающего её работу программного обеспечения. Для малых и средних организаций использование платных компонентов может перечеркнуть все достоинства собственной информационной системы, а выбирая свободно распространяемые программные продукты необходимо учитывать их надёжность, безопасность, дальнейшее развитие, наличие аналогов и возможность их использования при расширении информационной системы.

Анализируя данную информацию, можно сделать вывод, что потребность организации к информационной системе, может быть удовлетворена созданием собственной ИС.

Рекомендованное и общепризнанное сообществом разработчиков информационных систем решение, продиктованное здравым смыслом — это присутствие в организации централизованной системы управления базой данных, в которой будет храниться, обрабатываться и вноситься информация необходимая для работы организации. Кроме этого организация получает хорошую программную среду для дальнейшей работы с информацией иными средствами, программными комплексами. Основная особенность СУБД - это наличие процедур для ввода и хранения не только самих данных, но и описаний их структуры. Централизованное хранение обеспечивает такие преимущества, как целостность и непротиворечивость информации, исключение дублирования, обеспечение сохранности и безопасности данных. Также возможно создание ролевой модели работы с данными находящимися в централизованной системе управления базой данных.

Необходимость использования актуальной информации из информационной базы может появиться и вне офисной среды, к примеру, в служебной командировке или на деловой встрече для демонстрации роста продаж или каталога продукции. В рамках такой деятельности удобно иметь

возможность оперативно получить доступ к базе данных и средствам формирования отчётов, графиков, при этом имея стандартное аппаратное и программное обеспечение.

Современные возможности интернет технологий и централизованное хранение информации даёт и такие инструменты, как интеграция со сторонними информационными системами [4], например, в рекламе. В рекламной деятельности торговая организация может автоматизировать отправку данных о товарах и ценах в специализированные рекламные организации, предоставляющие услуги торговых интернет площадок, а также в собственные торговые информационные системы, такие как интернет-магазин, электронные рассылки клиентам.

Практические реализации программного обеспечения во всём мире множество раз сталкивались с проблемами совместимости, наиболее часто это выражалось в возможности использования того или иного программного продукта их пользователями, а также поддержкой программным продуктом иного ПО разработанного сторонними разработчиками. Решение проблем совместимости привело к разработке большого числа международных стандартов и соглашений в области применения и разработки информационных систем. Основополагающим понятием стало понятие открытые системы [8].

Открытую систему можно определить в качестве системы использующей согласованный набор международных стандартов на информационные технологии и профили функциональных стандартов, которые специфицируют интерфейсы, службы и поддерживающие их форматы, чтобы обеспечить взаимодействие и мобильность программных приложений, данных и персонала [8,10].

Это определение, сформулированное специалистами института IEEE (Institute of Electrical and Electronic Engineers), унифицирует содержание среды, которую предоставляет открытая система для широкого использования. В настоящее время общепризнанным координационным



центром (<https://www.oasis-open.org>) по разработке и согласованию стандартов открытых систем является OASIS (Organization for the Advancement of Structured Information Standards). В качестве спонсоров данного центра выступают такие IT-гиганты, как IBM и Microsoft.

Общие свойства открытых информационных систем можно описать следующим образом [14]:

мобильность/переносимость — такое свойство является полезным при необходимости переноса программ и данных в случае замены и модернизации аппаратного обеспечения информационной системы, а также обеспечивает возможность работы с ИС специалистов, без их переподготовки по случаю модификаций информационной системы;

расширяемость/масштабируемость — данное свойство должно упростить и обеспечить возможность добавления нового функционала в информационные системы или изменять уже имеющиеся, не затрагивая иные функциональные части информационной системы;

стандартизируемость — проектирование и разработка информационных систем должна осуществляться на основе согласованных международных стандартов и предложений, реализация открытости осуществляется на базе функциональных стандартов (профилей) в области информационных технологий;

взаимодействие — возможность взаимодействия с другими ИС и их интеграция является важным свойством, в этом способствует наличие технических средств, обеспечивающих реализацию объединения разных ИС при помощи сетевых технологий;

дружественность к пользователю — понятные, привычные и универсальные интерфейсы при взаимодействии «человек-машина» позволяют использовать информационную систему пользователям без прохождения специальной «компьютерной» подготовки.

Разработка и проектирование информационной системы для торговой организации с учётом перечисленных свойств позволит реализовать систему

«дружелюбную» к конечному пользователю, исключить такие недостатки как: асинхронность наполнения базы данных, дублирование информации, потеря актуальности, даст возможность интеграции с иными ИС и стандартизировать как компоненты, так и систему в целом.

Следует учесть и то, что возможность сделать информационную систему открытой системой можно лишь рассматривая перечисленные свойства как комплекс взаимосвязанных мер, принятых при проектировании, разработке и внедрении информационной системы. Таким образом, свойства будут являться не отдельными, а дополняющими и расширяющими друг друга, организуя открытую систему. Придерживаясь принципов открытых систем будут достигнуты поставленные цели и задачи, решены проблемы, с которыми сталкиваются как торговые, так и иные организации в процессе информатизации и автоматизации своей деятельности.

Анализируя примеры хранения, наполнения и обработки данных в информационных системах торговых организаций не имеющих свойств открытых систем или исключаяющих некоторые из них, можно выделить следующие возникающие проблемы:

- *отсутствие единого реестра (базы данных)* — при децентрализованном хранении и интерфейсе доступа к данным возникает ряд проблем: дублирование информации, потеря актуальности, затруднения с отчётностью.
- *Невозможность организации удалённого доступа* — вызывает сложность в своевременном получении данных, необходимость дополнительных процессов для передачи информации, потеря стратегического преимущества, связанного со временем получения данных.
- *Необходимость в специализированном клиентском ПО* — ограничивает конечного пользователя в выборе аппаратного и программного обеспечения для получения доступа к данным.

- *Невозможность или ограничение интеграции со сторонним ПО* — ограничивает организацию в функциональных возможностях информационной системы или связывает расширение и модернизацию ИС с определённым разработчиком/средой разработки.

Анализ литературы и практического опыта разработчиков интерфейсов для информационных систем, приводит к актуальному и популярному решению - создание web-приложения, которое позволит совместить комплекс свойств открытых систем.

Под определением **«Веб-приложение»** понимается прикладное программное обеспечение в реализации представляющее собой клиент-серверное решение, обмен информацией в котором происходит посредством сети (локальной или глобальной) через HTTP и/или HTTPS прикладные протоколы передачи гипертекста [6]. В данном решении клиентская часть может быть независимой от аппаратного и программного обеспечения, выполняя функцию пользовательского интерфейса (рис. 1.). Также следует упомянуть, что реализация клиентского интерфейса в веб-приложении может не требовать больших ресурсов к аппаратному обеспечению, выполняя ограниченное количество вычислений необходимых для отображения полученной информации из сервера и отправки необходимых запросов обратно.

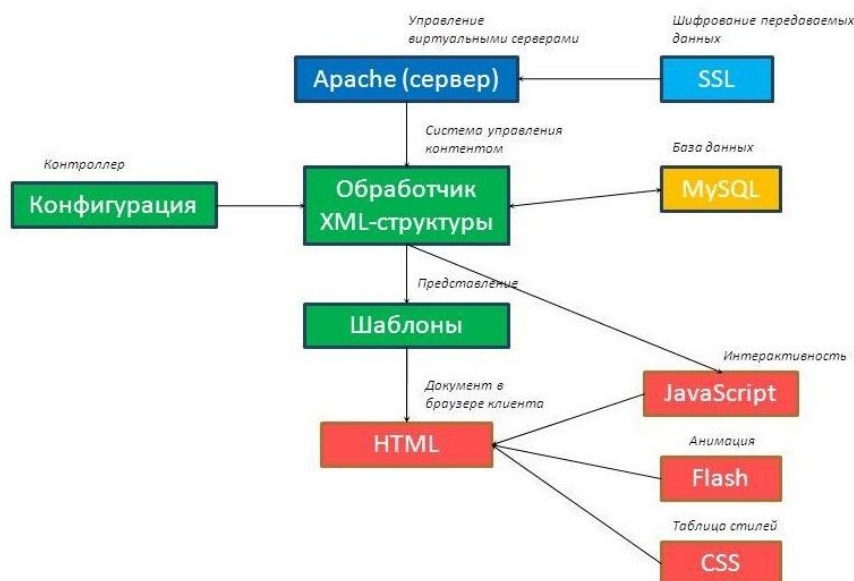


Рис. 1. Архитектура веб-приложения

**Веб-приложения имеют свои положительные особенности, соблюдающие и дополняющие свойства открытых систем [4]:**

- Клиентская часть выполняется независимо от операционной системы или аппаратного обеспечения клиента. В качестве программного обеспечения используются интернет-обозреватели (браузеры). Однако при разработке и тестировании клиентского интерфейса следует учесть, что приложение должно иметь свойства кроссбраузерности, из-за различной реализации браузерами стандартов HTML, CSS и DOM. Однако на практике большая часть стандартов ими соблюдена, а различия вытекают из-за разного времени применения новейших стандартов разработчиками этих программ обновлений.
- Веб-приложения представляют собой распределённую информационную систему из серверной и клиентской частей. Серверная часть проектируется для обеспечения необходимого количества клиентских обращений и имеет возможность масштабирования.
- В процессе разработки, веб-приложение может быть открыто для тестирования с удалённых компьютеров, что является положительным фактором для применения гибких методик разработки.
- Разработка веб-приложения является хорошей перспективой, особенно для торговых организаций, учитывая растущую и уже повсеместную сферу электронной коммерции.
- Предоставляются широкие возможности социального взаимодействия, взаимодействия и интеграции ИС с другими информационными системами функционирующими в глобальной сети.
- Благодаря работоспособности на любом аппаратном обеспечении и актуальности таких систем, веб-приложения могут составлять реальную конкуренцию нативным приложениям операционных систем (Apple iOS, Google Android, Windows Phone). В обозримом будущем конкурентоспособность может сильно возрасти благодаря работе W3C

(организация, разрабатывающая и внедряющая технологические стандарты для Всемирной паутины) над созданием открытых стандартов, дающих веб-приложениям доступ к аппаратной составляющей устройств, что сильно расширит возможности клиентской составляющей.

Как видно из перечисленных свойств и особенностей веб-приложений — они весьма подходят к определению открытых систем и соблюдают их принципы. Наиболее яркое осуществление свойств открытых систем — это внутреннее и внешнее взаимодействие веб-приложения посредством сетевых технологий, что является соблюдением свойства взаимодействия. Мобильность и переносимость отлично обеспечивается на клиентском уровне — как правило любая операционная система содержит браузер, который является программной основой для работы конечного пользователя с веб-приложением, на серверной стороне переносимость обеспечивается множеством совместимых аналогов программного обеспечения (такого как СУБД), возможностью запуска серверного ПО в различных операционных системах и на аппаратных решениях, лёгкость переноса серверной части на другое оборудование. Черта масштабируемости хорошо присутствует в веб-приложениях — отчасти это обеспечивается стандартизацией разработки и реализации веб-приложений, из этого также вытекает свойство стандартизации, хотя некоторые аспекты программирования имеют довольно молодые и не устоявшиеся стандарты, они весьма быстро распространяются благодаря сфере и области функционирования данных информационных систем и возможности быстрого внедрения. Дружественность клиентского интерфейса к пользователю во многом соблюдается благодаря наличию технологий и стандартов по разработке веб-приложений для различных устройств, часто этот фактор сводится к самому значительному: размер экрана и вид используемого аппаратного обеспечения и способ управления (классические клавиатура и компьютерная мышь или сенсорный экран). Веб-приложение позволяет сделать интерфейс удобным для конечного

пользователя независимо от того, использует тот смартфон (рис. 2.), персональный компьютер или телевизор с функцией SMART-TV. Клиентская часть веб-приложения может быть реализована в максимально простом и понятном виде, для решения конкретных задач пользователя «под ключ».

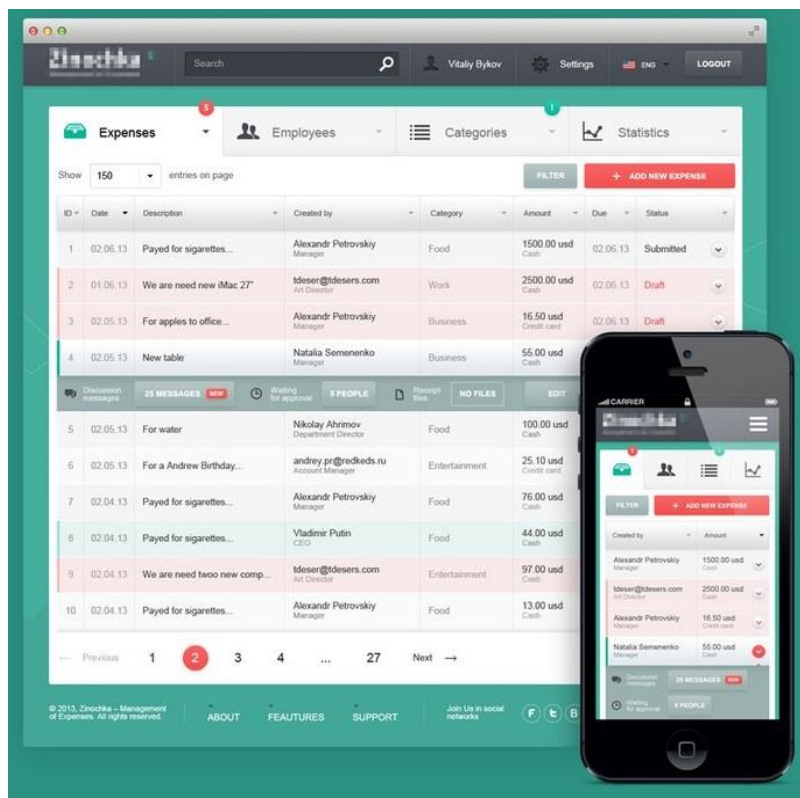


Рис. 2. Интерфейс веб-приложения на разных экранах

Организация разработки веб-приложения весьма гибкая, на конечный выбор методологии могут влиять такие факторы, как масштаб информационной системы, финансовые и временные рамки разработчика, техническое задание заказчика.

**Современные аспекты и концепции в методиках для разработки веб-приложений:**

- **Модель горизонтальной организации проектной группы** возникла из-за недостатков традиционной иерархической структуры в качестве эффективного аналога. Такая группа — это многопрофильная, небольшая команда, каждый в которой принимает на себя ответственность за ту или иную часть разработки, а также дополняющий компетентность смежных членов команды, что

обеспечивает единое видение проекта каждым участником. В таких командах выставляются высокие требования к компетенции и конечному результату, поощряется желание в образовании совершенствовании навыков. Не редко такие команды собираются в крупных организациях для разработки конкретной информационной системы (пример, Microsoft Solution Framework), однако, есть и антонимичные примеры — экстенсивная модель роста за счёт увеличения прибыли, путём повышения набора низкооплачиваемых специалистов, что вызывает постоянную текучку кадров, но снижает риски из-за кадровых потерь.

- **Модель проектной группы** включает в свой состав специалистов, исполняющих роли: дизайнера/проектировщика интерфейсов, разработчика, специалиста по оптимизации, тестера, управляющего проектом. Все участники в такой проектной группе равноправны, каждый участник исполняет определённую роль или несколько ролей сразу, а управляющий проектом является техническим специалистом. Минимальный коллектив должен состоять не менее, чем из 3 человек.
- **Интеграция IT специалисту функций веб-разработчика** является созданием универсального специалиста, который должен иметь высокое качество навыков в области разработки как серверной составляющей, так и клиентской части — это вёрстка, проектирование баз данных, оптимизация и проектирование интерфейсов. Специалист может быть развит больше в одной области, чем в другой, но должен быть способен реализовать проект целиком, пусть и за большее время, чем это сделает команда специалистов. Современные требования к разработчикам веб-приложений предполагают широкое видение разработки проекта за рамками своей прямой компетенции, т.к. различные части веб-приложения во многом дополняют друг друга и плотно взаимодействуют.

- **Жизненный цикл разработки веб-приложения** схож с методологией Rational Unified Process и состоит из 4 фаз, которая может быть исполнена только один раз или повторяться несколько раз при необходимости. Фаза «Начало» формирует видение и границы проект, определяет основные требования, ограничения и ключевую функциональность продукта. Также при этой фазе оцениваются риски. В фазе «Уточнение» производится анализ предметной области и построение архитектуры приложения, разработка интерфейсов и дизайна. В фазе «Конструирование» происходит реализация большей части функциональности. В фазе «Внедрение» - происходит тестирование и оценка, определяется качество получившегося веб-приложения, проводится оптимизация.
- **Критичный подход к использованию и выбору готовых решений.** Само использование существующих программных решений для повышения скорости и/или качества приложения может являться хорошим фактором, но при этом необходимо учитывать влияние таких решений на безопасность, архитектуру и дальнейшую масштабируемость. При использовании готовых решений должно быть понимание их логики работы и возможность модификации. Поэтому для серьёзных проектов требовательных в первую очередь к безопасности не используются полностью готовые системы управления веб-содержимым (WordPress, Joomla, OpenCart).
- **Заказчик всегда рядом** — концепция экстремального программирования. Заказчик представляет собой конечного пользователя продукта, который должен быть всегда на связи для диалога.
- **Парное программирование** — это техника, ставшая весьма продуктивной и полезной при разработке веб-приложений с определёнными условиями. Парное программирование производится двумя специалистами, из которых один осуществляет кодирование в



деталей, а другой сосредоточен на картине в целом. Через некоторое время происходит смена ролей. Смежный принцип коллективного владения кодом означает, что каждый разработчик несет ответственность за весь исходный код.

Исходя из вышеперечисленных способов разработки веб-приложений можно сделать вывод, что методика разработки может быть разной при разных проектах, а также совмещать в себе лучшие качества, организуя гибридные варианты в процессе реализации конкретных задач. На выбор методики может влиять время выполнения, обоснование используемых ресурсов и требовательность к конечному функционалу.

## 1.2. Стандарты разработки веб-приложений

Кроме методики организации разработки веб-приложений, не менее важными являются правила написания кода. Стандартизация написания кода веб-приложений — это развивающиеся решения, одной из популярных и глобально применяющихся методик, является **БЭМ-методология** (Блок, Элемент, Модификатор), которая включает в себя компонентный подход к веб-разработке, подразумевающий разделение кода и файловой структуры интерфейса на независимые блоки. Данная методология разработана российской компанией Яндекс и описывает подход к написанию кода всех веб-технологий для построения интерфейса (HTML, CSS, JavaScript), а также составление документации, проведение тестов и т.д.

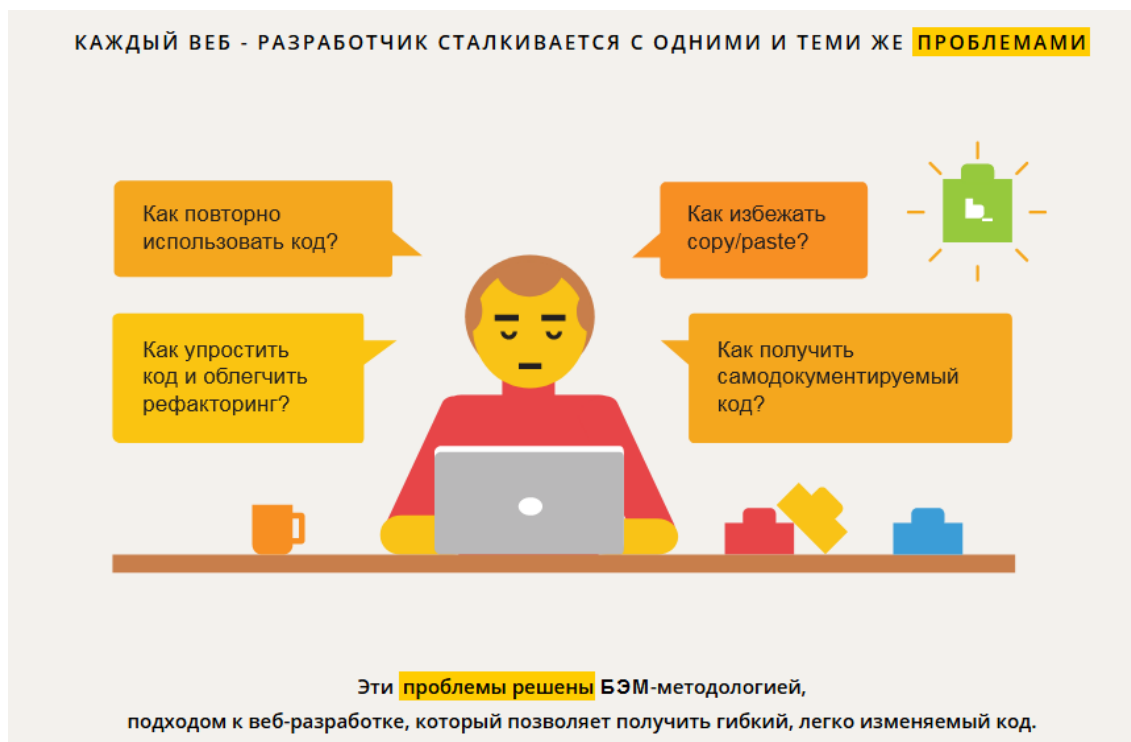


Рис. 3. БЭМ-методология как решение частых проблем веб-разработки

БЭМ-методология сопровождается большим количеством технической документации, готовыми к работе примерами, инструментариями, доступными в соответствующем разделе сайта «Технологии Яндекс» (<https://tech.yandex.ru/bem/>).

*БЭМ включает в себя:*

- Методологические рекомендации - это простые советы, помогающие организовать проект, наиболее быстро его реализовать и удобно поддерживать длительное время.
- Технологии и библиотеки с открытым исходным кодом — готовая реализация рекомендаций БЭМ.
- Инструменты для автоматизации работы с методологией БЭМ.

*Преимущества БЭМ:*

- Простая поддержка структуры кода при росте проекта.
- Повторное использование кода — позволяет сократить сроки и стоимости новых разработок за счёт использования одинаковых компонентов.

- Точечные изменения с минимальными затратами: обновление дизайна, добавление функциональных элементов и т. д.
- Масштабирование — код развивается по заранее известным правилам.
- Общая терминология — позволяет упростить взаимодействие разных разработчиков.
- Автоматизация — общие единые правила способствуют автоматизации процессов. Часть кода может быть сгенерирована автоматически.

БЭМ рекомендуют многие популярные разработчики, консультанты и технические директора. Его используют в таких IT-гигантах как Google, Яндекс, BBC, Alpha-Bank, BuzzFeed, ЕРАМ. Также БЭМ можно использовать не только в веб-разработке, но и в любом другом языке программирования и фреймворке, т. к. методология предлагает только абстрактные практики по повышению надёжности и повторному использованию кода [23].

Кроме стандартизированной разработки веб-приложения в части интерфейса, необходимо также стандартизировать разработку серверной части, основой которого является PHP-код. Наиболее популярным и применяемым в мире является **стандарт кодирования PSR**. Proposing a Standards Recommendation — предлагаемые рекомендации стандартов, которые в свою очередь разделены на несколько независимых блоков. Данный стандарт создан группой взаимодействия фреймворков (PHP-FIG), является развивающимся и постоянно дополняющимся, охватывая всё больше аспектов PHP-программирования [28]. На сайте сообщества можно найти как утверждённые рекомендации, так и рекомендации, находящиеся в разработке (<http://www.php-fig.org>).

**Описание существующих на данный момент PSR стандартов [28,34]:**

**PSR-1 - основной стандарт кодирования.** Данный стандарт описывает наиболее общие правила, такие как использование тегов PHP, кодировка файлов, разделение места объявления функции, класса и места их использования, а также именование классов и методов.

**PSR-2 - руководство по стилю кода.** Расширяет первый стандарт, регулирует такие вопросы использования в коде как «табуляция или пробелы», переводы строк, максимальная длина строк, правила оформления управляющих конструкций (рис. 4.) и другое.

**PSR-3 - интерфейс протоколирования.** Этот стандарт разработан для того, чтобы обеспечить логирование (журналирование) в приложениях, написанных на PHP.

**PSR-4 - стандарт автозагрузки.** Это, наверное, самый важный и нужный стандарт PSR. Классы, которые реализуют PSR-4, могут быть загружены единым автозагрузчиком, что позволяет частям и компонентам из одного фреймворка или библиотеки быть использованными в других проектах.

**PSR-6 - интерфейс кеширования.** Кэширование используется для повышения производительности системы. И PSR-6 позволяет стандартно сохранять и извлекать данные из кэша, используя унифицированный интерфейс.

**PSR-7 - интерфейс HTTP-сообщений.** При написании сложных веб-приложений на PHP, почти всегда приходится работать с HTTP заголовками. Конечно, язык PHP предоставляет нам уже готовые возможности для работы с ними, такие как суперглобальный массив `$_SERVER`, функции `header()`, `setcookie()` и т.д., однако их ручной разбор чреват ошибками, да и не всегда можно учесть все нюансы работы с ними. И вот, чтобы облегчить работу разработчику, а также сделать единообразным интерфейс взаимодействия с HTTP протоколом был принят данный стандарт.

**PSR-11 - интерфейс контейнера.** При написании PHP программы часто приходится использовать сторонние компоненты. И чтобы не заблудиться в обилии зависимостей были придуманы различные методы управления зависимостями кода, зачастую несовместимые между собой, которые данный стандарт и приводит к общему знаменателю.

**PSR-13 - гипермедиа ссылки.** Данный интерфейс призван облегчить разработку и использование прикладных программных интерфейсов (API).

**PSR-16 - интерфейс простого кэширования.** Является продолжением и улучшением стандарта PSR-6.

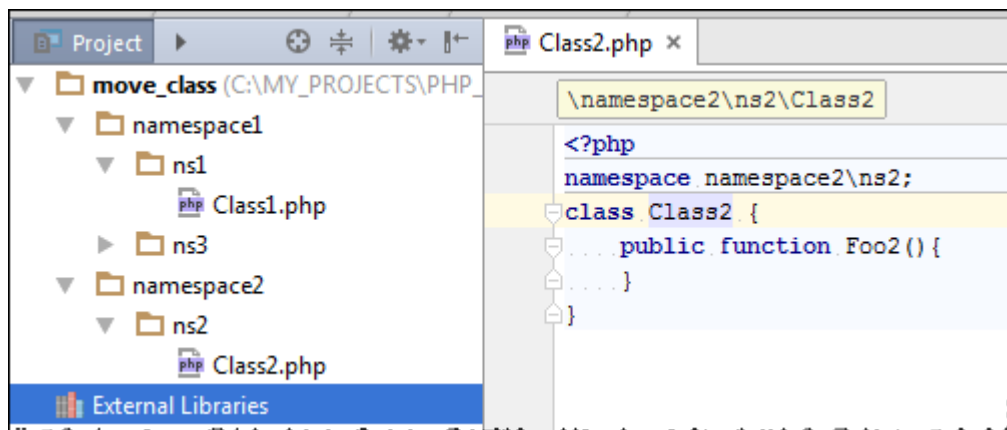


Рис. 4. Оформление кода и файловая структура в соответствии с PSR

**Консорциум Всемирной паутины** (*World Wide Web Consortium, W3C*) — это организация, разрабатывающая и внедряющая для Интернета единые принципы и технологические стандарты (называемые рекомендациями), которые затем внедряются производителями программ и оборудования. Консорциум возглавляет сэр Тимоти Джон Бернерс-Ли, автор множества разработок в области информационных технологий [40].

Стандарты, рекомендуемые W3C, являются открытыми и свободными для внедрения кем угодно, без финансовых отчислений консорциуму. Организация не проводит сертификацию на соответствие стандартам и не планирует этим заниматься. Благодаря этому рекомендации консорциума получили широкое распространение, однако это вызывает проблему с тем, что некоторые производители следуют рекомендациям лишь частично (такое внедрение не нарушает общих стандартов W3C). Также внедрение разделено на несколько степеней внедрения — каждый сам решает в какой степени необходимо внедрение W3C стандартов. Из практических примеров — это частичное внедрение новых стандартов W3C в работе веб-обозревателей, что обуславливает веб-разработчиков необходимостью проводить дополнительное тестирование на кроссбраузерность веб-приложения.

Стандарты W3C имеют черту хорошо детализированных рекомендаций и доступны как начинающим веб-мастерам, так и экспертам-программистам. Технические спецификации консорциума сопровождаются обилием руководств и советов, которые облегчают понимание и внедрение стандартов.

В последнее время W3C проводит много работы по популяризации своих стандартов, их совершенствованию и расширению.



Рис. 5. Один из популярнейших стандартов W3C – это HTML версии 5

#### **Принципы и концепции, утверждённые W3C:**

- **Общедоступность** – при реализации данного принципа, большое внимание уделяется доступности ресурсов глобальной сети для людей с ограниченными физическими возможностями.
- **Аппаратная независимость** - этот принцип утверждает, что ресурсы и протоколы не должны быть кроссплатформенными, т.е. рассчитаны работать на любом аппаратном обеспечении.
- **Интернационализация** – прежде всего подразумевает языковой барьер и призывает к доступности информации на всех языках мира, возможности и равноправию использования национальных шрифтов в документах и системах разработанных на основе стандартов.
- **Патентная политика** - здесь речь идёт о патентовании рекомендаций W3C таким образом, чтобы их можно было внедрять бесплатно, на основе лицензии W3C Royalty-Free License.
- **Многоформенное взаимодействие** – принцип, предполагающий наличие различных форм и режимов взаимодействия программы с человеком.

- **Семантическая паутина** – свежая и актуальная концепция, призывающая размещать информацию в более понятном для компьютеров виде. Семантическая паутина сейчас реализуется как надстройка над существующей Всемирной паутиной.
- **Мобильная паутина** - этот принцип определяет, что ресурсы Всемирной сети должны быть доступны для мобильных устройств.

Одни из самых используемых стандартов, утверждённых W3C: CSS, DOM, HTML, HTTP, OWL, PNG, RDF, SOAP/XMLP, Style, SVG, URI/URL, XML.

Применение данных стандартов и методологий позволит создать информационную систему, соответствующую современным технологическим требованиям и имеющую высокую конкурентоспособность.

## Глава 2. Проектирование и создание информационной системы для торговой организации

### 2.1. Разработка серверной части информационной системы

Под серверной частью подразумевается back-end составляющая web-приложения. Back-end и front-end — это термины в программной инженерии, которые различают согласно принципу разделения ответственности между внешним представлением и внутренней реализацией соответственно. Front-end — интерфейс взаимодействия между пользователем и основной программно-аппаратной частью (back-end). Front-end и back-end могут быть распределены между одной или несколькими системами [19].

Серверная, back-end часть информационной системы принимает и обрабатывает клиентские запросы, обращается к СУБД при необходимости, формирует и отправляет ответ клиенту.

Серверная часть состоит из аппаратного и программного обеспечения. В качестве аппаратного обеспечения у компании ISP SERVER (<https://ispserver.ru>) был арендован виртуальный выделенный сервер с рабочими характеристиками: 1GB RAM, 100GB HDD, 2 GHz CPU и предустановленной операционной системой FreeBSD.

**Виртуальный сервер (VPS)** — это услуга, в рамках которой пользователю предоставляется часть ресурсов одного физического сервера. В плане управления, такой сервер практически не отличается от физического, на него можно установить полноценную операционную систему, и он имеет один или несколько статических IP-адресов для выхода в интернет.

**FreeBSD** — это свободная операционная система семейства UNIX (<https://www.freebsd.org/ru/>). ОС распространяется под лицензией BSD и является популярной системой для сетевых серверов (управление сетью и различные сетевые службы).



При помощи системы портов (технология установки программного обеспечения сторонних разработчиков в операционной системе FreeBSD) были установлены необходимые компоненты информационной системы, обеспечивающие работу сервера в роли веб-сервера и тем самым веб-приложения: система управления базами данных, HTTP-сервер и PHP-интерпретатор.

Активная деятельность по отысканию приемлемых способов обобществления непрерывно растущего объема информации привела к созданию в начале 60-х годов специальных программных комплексов, называемых "Системы управления базами данных" (СУБД).

Основная особенность СУБД - это наличие процедур для ввода и хранения не только самих данных, но и описаний их структуры. Файлы, снабженные описанием хранимых в них данных и находящиеся под управлением СУБД, стали называть банками данных, а затем "Базами данных" (БД) [20].

Выбор СУБД — это выбор из числа многочисленных программно-лингвистических средств общего и специального назначения, обеспечивающих управление созданием и использованию баз данных. В системе управления базой данных для торговой организации планируется хранить максимальное количество информации о текущей бизнес-деятельности, начиная от каталога товаров и заканчивая данными о заказах клиентов. При выборе были учтены функциональные возможности, документальная поддержка, наличие литературы, масштабируемость и интеграция с другими программными продуктами. Немаловажными стали и такие параметры как возможности системы для работы, как с малым количеством данных, так и возможность «расти» вместе с организацией. Малому бизнесу могут потребоваться только ограниченные возможности, т.е. небольшое количество информации, но требования могут существенно расти в течение времени, и переход на другую систему управления базами данных может стать проблемой. Учтены и стоимость использования СУБД, наличие

на рынке труда узких специалистов работающих с данной СУБД в случае потребности их найма, наличие достаточного количества технической документации и литературы.

При выборе были рассмотрены наиболее популярные и развитые системы управления базой данных [16,18].

## Oracle Database

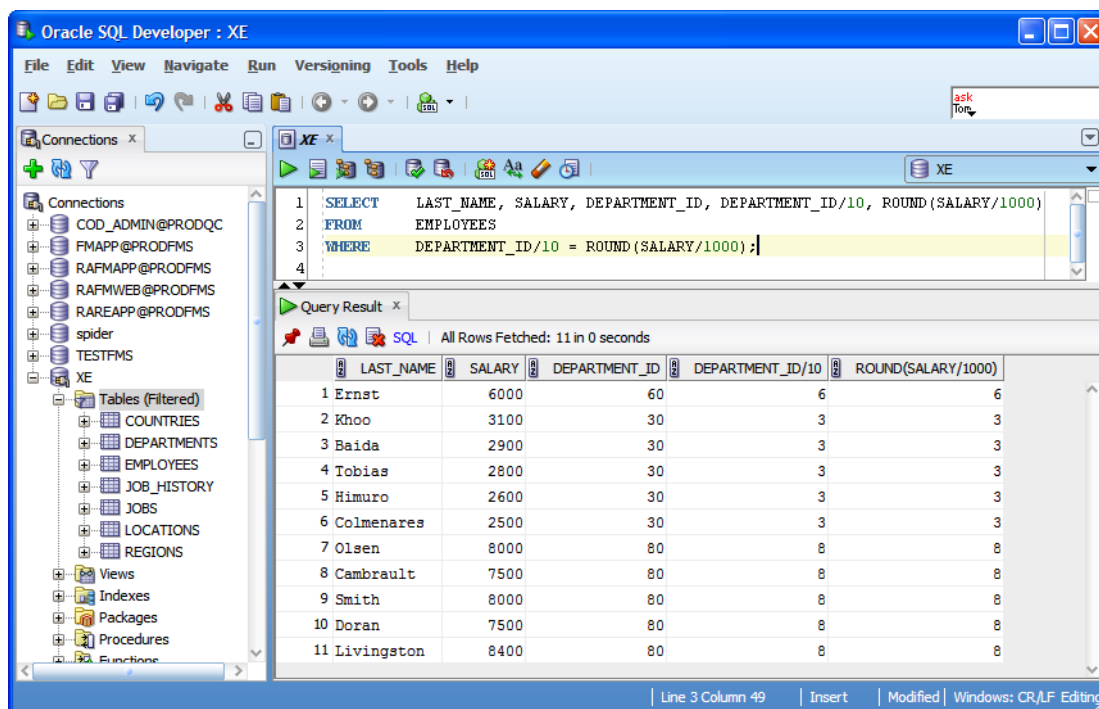
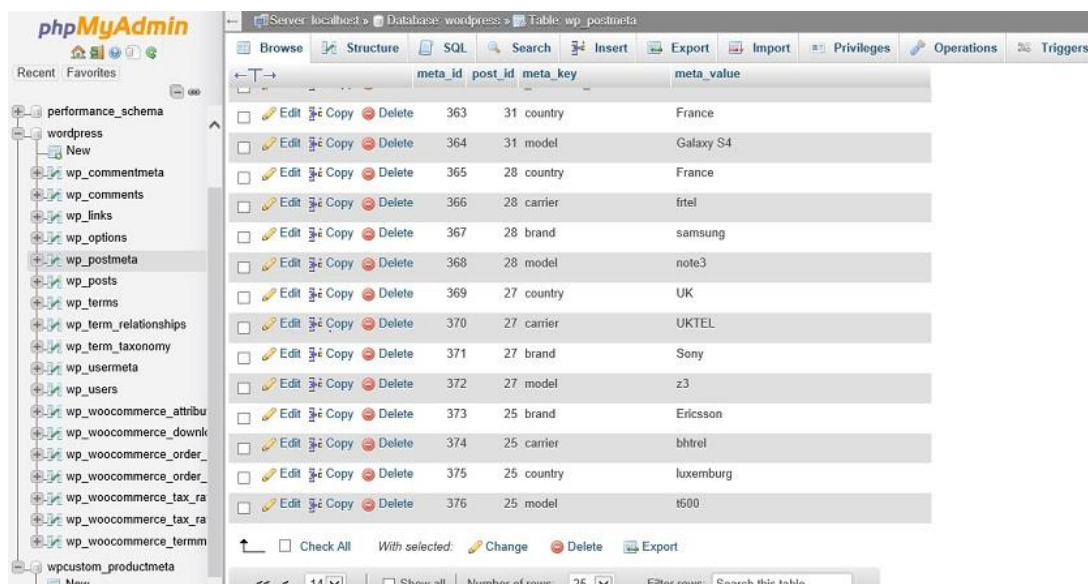


Рис. 6. Базы данных и таблицы в Oracle SQL Developer

Одна из популярных и современных объектно-реляционных система управления базами данных от компании Oracle. Первая версия системы была создана в 70-х годах, существует несколько различных версия для удовлетворения узконаправленных потребностей в информатизации. Последняя из выпущенных версий предназначена для облачных сред, т.е. размещается на 1 и более серверах, позволяет хранить и работать с очень большими объёмами записей. Присутствует сильная безопасность, обусловленная изоляцией каждой транзакцией в запросах к базе данных. Данная база данных имеет инновационные особенности, широкий набор инструментов и задаёт планку качества и технологичности для многих других СУБД. Oracle Database является платным продуктом, цена для малых и развивающихся организаций весьма высока, также система требует

значительно лучшее аппаратное обеспечение в плане ресурсов. На российском рынке труда не много специалистов работающих с данной БД. Техническая документация и ссылки для скачивания доступны на официальном сайте: <https://www.oracle.com>.

## MySQL



	meta_id	post_id	meta_key	meta_value
<input type="checkbox"/>	363	31	country	France
<input type="checkbox"/>	364	31	model	Galaxy S4
<input type="checkbox"/>	365	28	country	France
<input type="checkbox"/>	366	28	carrier	frtel
<input type="checkbox"/>	367	28	brand	samsung
<input type="checkbox"/>	368	28	model	note3
<input type="checkbox"/>	369	27	country	UK
<input type="checkbox"/>	370	27	carrier	UKTEL
<input type="checkbox"/>	371	27	brand	Sony
<input type="checkbox"/>	372	27	model	z3
<input type="checkbox"/>	373	25	brand	Ericsson
<input type="checkbox"/>	374	25	carrier	bhtrel
<input type="checkbox"/>	375	25	country	luxemburg
<input type="checkbox"/>	376	25	model	t600

Рис. 7. Базы и таблицы MySQL в интерфейсе PhpMyAdmin.

Свободная реляционная СУБД от корпорации Oracle. Имеет большую популярность при разработке веб-приложений. Разрабатывается с 90-х годов, активно разрабатывается и поддерживается компанией разработчиком. Встроенные инструменты для работы с MySQL имеют многие решения в области программного обеспечения для веб-приложений. Является быстрой, надёжной и относительно использования - безопасной. В качестве аппаратного обеспечения может использоваться ЭВМ с довольно низкими ресурсами. Имеет как бесплатную версию под лицензией GNU GPL 2, так и платную с официальной поддержкой. На рынке труда имеется множество специалистов работающих с данной СУБД, а также большое кол-во технической документации, литературы, практических примеров использования и готовых продуктов в составе которых используется MySQL. Имеет возможности распределения нагрузки на несколько серверов. Техническая документация и ссылки для скачивания доступны на официальном сайте: <https://www.oracle.com>.

## Microsoft SQL Server

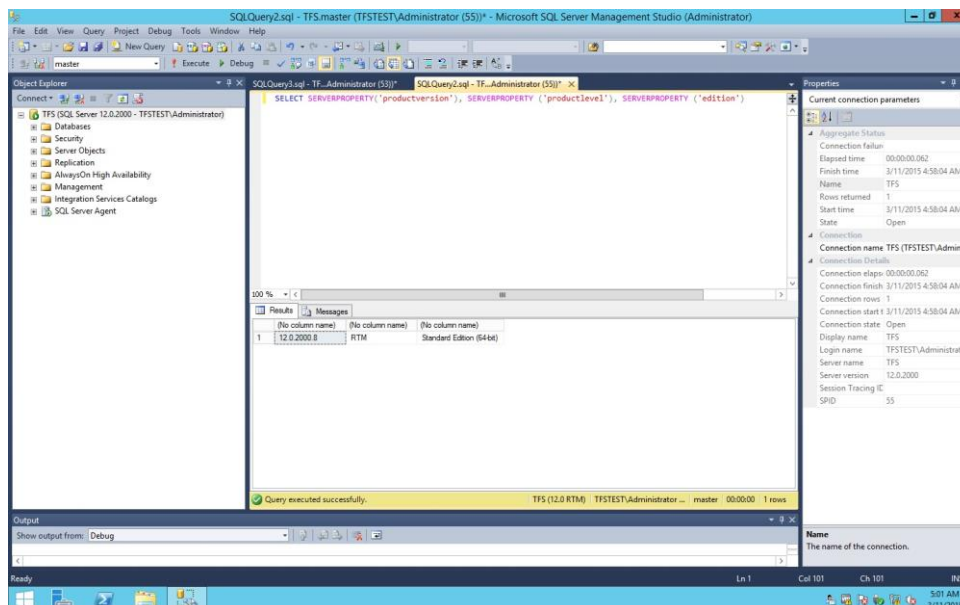


Рис. 8. SQL Server Manager

Система управления реляционными базами данных, разработанная корпорацией Microsoft. Ядро данной системы может предназначено для работы как на локальных серверах, так и на облачных. В 2016 году стала доступна в операционной системе Linux. Одной из уникальных особенностей версии 2016 года является временная поддержка данных, которая позволяет отслеживать изменения данных с течением времени. Последняя версия Microsoft SQL-сервер также позволяет совершать динамическую маскировку данных, которая гарантирует то, что только авторизованные пользователи будут видеть конфиденциальные данные. SQL Server является быстрым и стабильным средством, имеет хорошую техническую документационную основу и множество литературы, большое количество технических специалистов на рынке труда. Данная СУБД является платным решением, со средним ценником, хорошо ориентирована на работу с иными продуктами Microsoft. Техническая документация и ссылки для скачивания доступны на официальном сайте: <https://www.microsoft.com/ru-ru/sql-server/sql-server-2016>.

## PostgreSQL

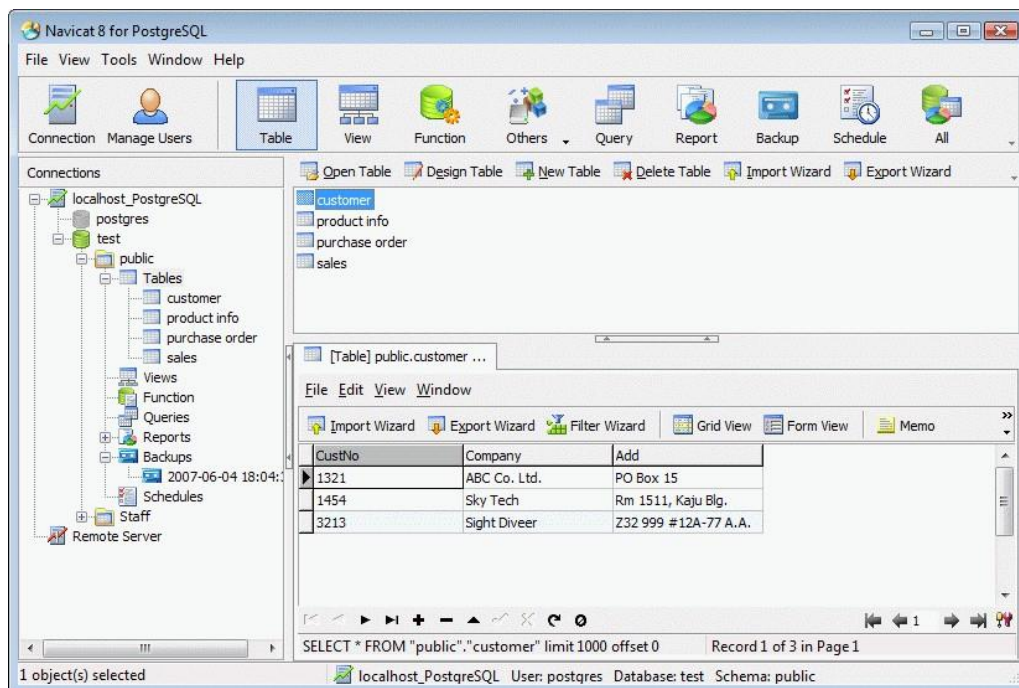


Рис. 9. Postgre база данных в интерфейсе программы Navicat 8

Свободная объектно-реляционная система управления базами данных, базирующаяся на языке SQL, распространяемая под лицензией License PostgreSQL . Может использоваться для обработки больших данных, хорошо масштабируема. Разрабатывается и поддерживается одноимённым сообществом и популярен среди веб-баз данных. Имеется достаточное кол-во литературы и специалистов, не требовательна к аппаратным ресурсам и хорошо подходит для проведения большого кол-ва запросов. Техническая документация и ссылки для скачивания доступны на официальном сайте: <https://www.postgresql.org>.

## Mariadb

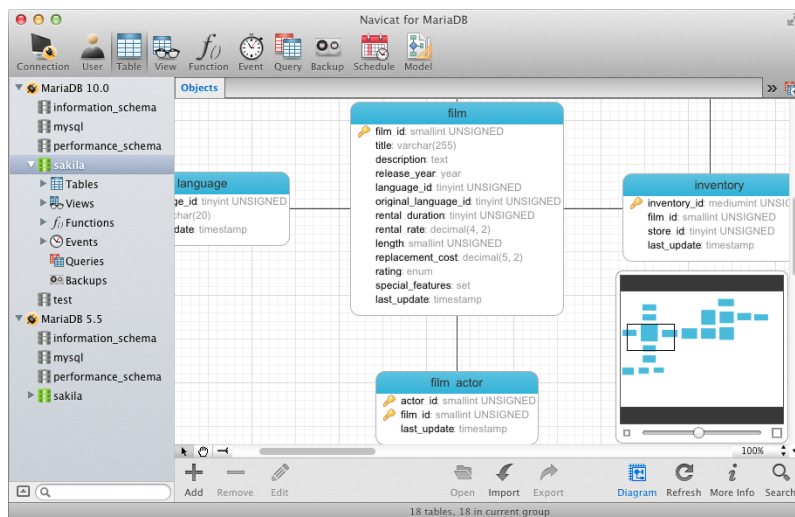


Рис. 10. База данных MariaDB в интерфейсе программы Navicat

Ещё одна свободно распространяемая система управления базой данных от компании MariaDB Corporation Ab и фонда MariaDB Foundation, распространяется под лицензией GNU GPL. Система является ответвлением от СУБД MySQL, разрабатывается первоначальными разработчиками MySQL с 2009 года. Система имеет и платный вариант с расширенной поддержкой и функционалом. Mariadb - это одна из самых быстрорастущих СУБД со множеством плагинов, однако ещё не имеющая широкой поддержки сторонним программным обеспечением, как другие СУБД и количеством литературы на русском языке, однако является полностью совместимым с MySQL продуктом. Является довольно безопасным и быстрым продуктом, однако стабильность ещё не так высока, как у СУБД уже давно присутствующих на рынке. Часто рассматривается как вариант альтернативный СУБД MySQL. Техническая документация и ссылки для скачивания доступны на официальном сайте: <https://mariadb.org>.



## DB2

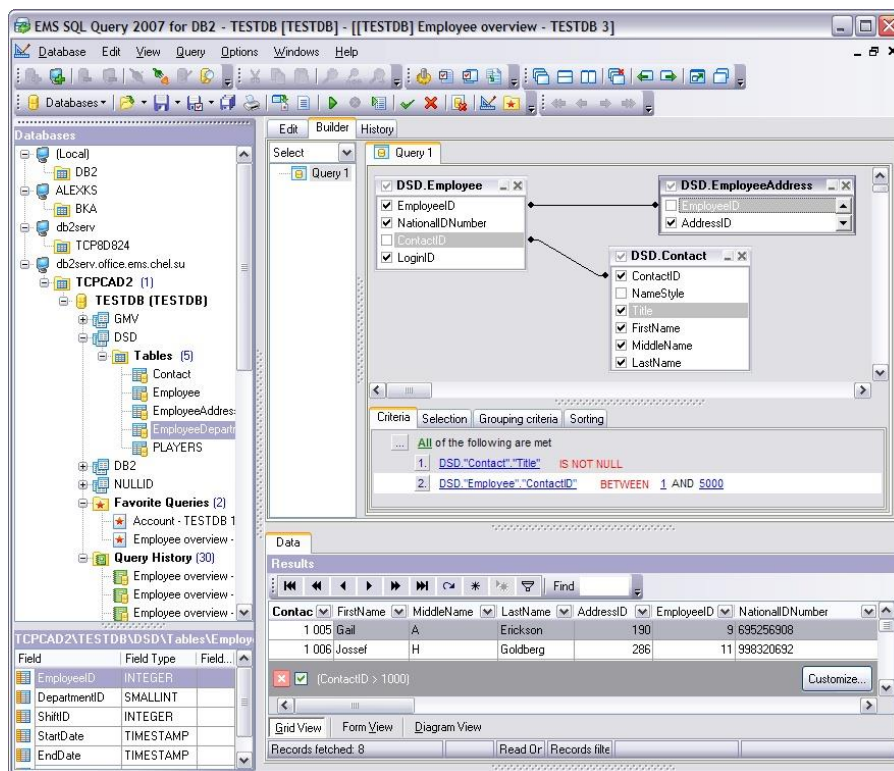


Рис. 11. База данных DB2 в интерфейсе EMS SQL Query 2007

Платная СУБД от компании IBM, имеющая возможности `posql`, читающая `json` и `xml`-файлы. Является кроссплатформенным решением, имеет множество документации и литературы, может быть размещено в облачном хранилище, на физическом сервере, или же и там, и там одновременно. Базовая поддержка доступна только в течение трех лет; после этого, вы должны заплатить за это. Подходит для: крупных организаций, которые делают имеют много ресурсов и обрабатывают большие базы данных. Техническая документация и ссылки для скачивания доступны на официальном сайте: <https://www.ibm.com/analytics/us/en/db2/>.

Из рассмотренных СУБД для решения задач торговой организации была выбрана MySQL. MySQL разработал Михаэль Видениус (Michael Widenius, [monty@analytikerna.se](mailto:monty@analytikerna.se)), она является небольшой и быстрой реляционной СУБД, основанной на традициях Hughes Technologies Mini SQL (`mSQL`).

Реляционные модели СУБД характеризуются простотой структуры данных, устранением избыточности, удобным для пользователя табличным

представлением и возможностью использования формального аппарата алгебры отношений и реляционного исчисления для обработки данных. Реляционная модель ориентирована на организацию данных в виде двумерных таблиц. Каждая реляционная таблица представляет собой двумерный массив и обладает следующими свойствами.

СУБД MySQL предоставляет следующие преимущества при её выборе [20]:

- реляционная модель данных;
- свободно-распространяемая под лицензией GNU GPL 2;
- хорошо адаптирована для веб-приложений;
- имеет хорошую базу технической документации и литературы;
- является активно развивающимся продуктом с большим опытом;
- большое количество специалистов на рынке труда;
- множество плагинов и отличная интеграция в другое программное обеспечение для веб-приложений;
- низкие требования к ресурсам аппаратного обеспечения.

Важный этап проектирования и разработки базы данных — это создание инфологической и даталогической моделей.

Под **инфологической моделью** понимают описание предметной области, выполненное с использованием специальных языковых средств, не зависящих от используемых в дальнейшем программных средств (это по существу блок-схема алгоритма создания базы данных). Цель инфологического моделирования - обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты).

#### **Требования к инфологической модели:**

- адекватное отображение предметной области;
- непротиворечивость;



- должна отражать взгляды и потребности всех пользователей системы;
- однозначная трактовка моделей;
- модель должна быть конечной;
- модель должна быть легко расширяемой, то есть иметь возможность ввода новых (удаления) данных без изменения ранее определенных;
- должна обладать свойствами композиции и декомпозиции (укреплять базу данных или расщеплять);
- должна быть легко реализуемой на ЭВМ;
- должна быть независимой от оборудования и языков организации базы данных на ЭВМ.

**Даталогическое проектирование** — это создание модели, являющейся моделью логического уровня и представляющей собой отражение логических связей между элементами данных. Эта модель строится в единицах допустимых конкретной СУБД. Описание логической структуры с помощью средств СУБД называется схемой. Так как это осуществляется с помощью конкретной СУБД, то модели должны быть описаны на языке описания данных этой СУБД. Такое описание и называют даталогической моделью данных.

База данных состоит из следующих таблиц:

- Таблица «Клиенты» - содержит информацию о частных и юридических лицах, которым реализуется товар.
- Таблица «Избранные клиенты» - содержит информацию о том, у каких пользователей какие клиенты добавлены в качестве избранных.
- Таблица «Поставщики» - содержит информацию об организациях, осуществляющих поставку продукции.
- Таблица «Заказы» - заказы, их статус и данные.
- Таблица «Товары» - продукт, реализуемый компанией и его свойства.
- Таблица «Бренды» - реестр брендов товаров и информации о них.
- Таблица «Группы» - группы товаров.

- Таблица «Конфиг» - содержит основные настройки ИС.
- Таблица «Пользователи» - пользователи системы, их свойства и права.
- Таблица «Логи» - содержит логи основных действий пользователей.
- Таблица «Страницы» - служебная информация о страницах/модулях ИС.
- Таблица «Курс валюты» - таблица хранит актуальный курс валют для вывода на главные страницы ИС и графики.
- Таблица «Чат» - содержит сообщения чата.
- Таблица «Задачи» - в таблице находятся все данные для осуществления функционала «Задачи».

Для составления даталогической (рис. 13.) и инфологической (рис. 12.) моделей базы данных, было использовано бесплатное специализированное веб-приложение DbDesigner (<http://dbdesigner.net>).

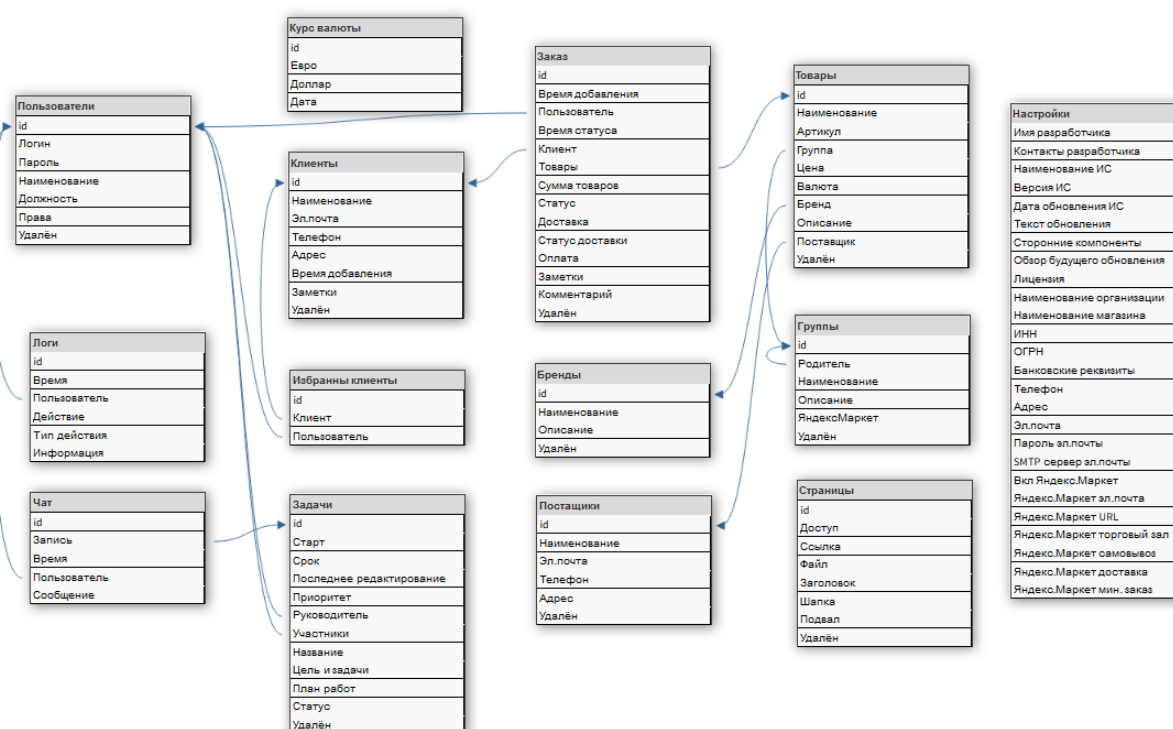


Рис. 12. Инфологическая модель базы данных

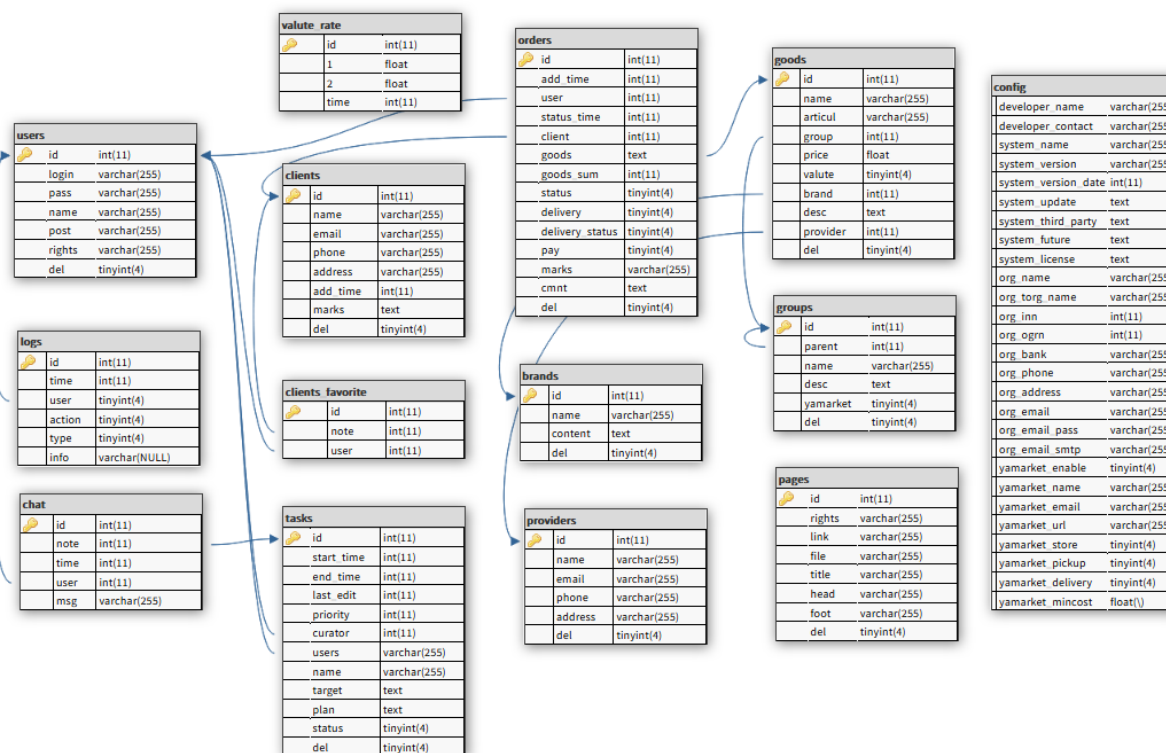


Рис. 13. Даталогическая модель базы данных

Как видно из спроектированных моделей базы данных — все таблицы некоторым образом связаны друг с другом. К примеру, в таблицы «Заказы» и «Избранные клиенты» имеют поля, связывающие их с таблицей «Клиенты» и хотя таблица «Заказы» никак не связывает себя напрямую с таблицей «Избранные клиенты», через общую для них таблицу «Клиенты» выстраивается характерная для реляционных СУБД связь. Таким образом связываются между собой и все остальные таблицы.

Таким образом, создана уже ставшая традиционной реляционная база данных, которая имеет зависимости между данными, но при этом избегает избыточности. Если системе необходимо показать заказы, то оно запрашивает данные о них из таблицы «Заказы» при этом, не запрашивая список клиентов и список товаров, что можно сделать при просмотре конкретного заказа, но и в этом случае запрашивать из базы данных придётся не все товары и всех клиентов, а только определённых, относящихся к данному заказу.

В качестве программного обеспечения на сервер были установлены СУБД mysql-server и mysql-client (<https://dev.mysql.com/downloads/mysql/>)

(рис. 14.) в качестве простого и быстрого доступа к управлению СУБД. При помощи MySQL клиента (рис. 14.) и запросов к СУБД были созданы таблицы с полями определённых типов, указаны уникальные индексы (id), которым было присвоено свойство «AUTO\_INCREMENT», указывающее, что порядковый номер будет генерироваться автоматически (предыдущее значение + 1) по мере добавления новых строк в таблицу.

```
root@s:~ # mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 194131
Server version: 5.7.17-log Source distribution

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

root@localhost [(none)]> use urgpu;
Database changed
root@localhost [urgpu]> select `phone`,`email` from `clients` where `id`='9';
+-----+-----+
| phone      | email      |
+-----+-----+
| 8 800 2000 500 | grand@example.ru |
+-----+-----+
1 row in set (0.00 sec)

root@localhost [urgpu]>
```

Рис. 14. Работа с СУБД MySQL через консоль MySQL-Client

Таким образом, разработана база данных торговой организации — основной компонент системы. Данная БД позволяет хранить, обрабатывать и дополнять информацию организации в едином месте.

Следующим шагом по настройке back-end части – стала установка HTTP-сервера, умеющего интерпретировать PHP-код.

**Apache HTTP Server** – это свободно распространяемый HTTP сервер под лицензией Apache License 2.0 (<http://www.apache.org/>). Он осуществляет 2 главные функции: взаимодействует с клиентом по HTTP протоколу и компилирует php-код при помощи соответствующего встроенного модуля (рис. 15.). Данная программа является наиболее актуальной и используемой для создания web-серверов [2]. Также Apache кроссплатформенный, надёжный и гибкий в конфигурации [30].

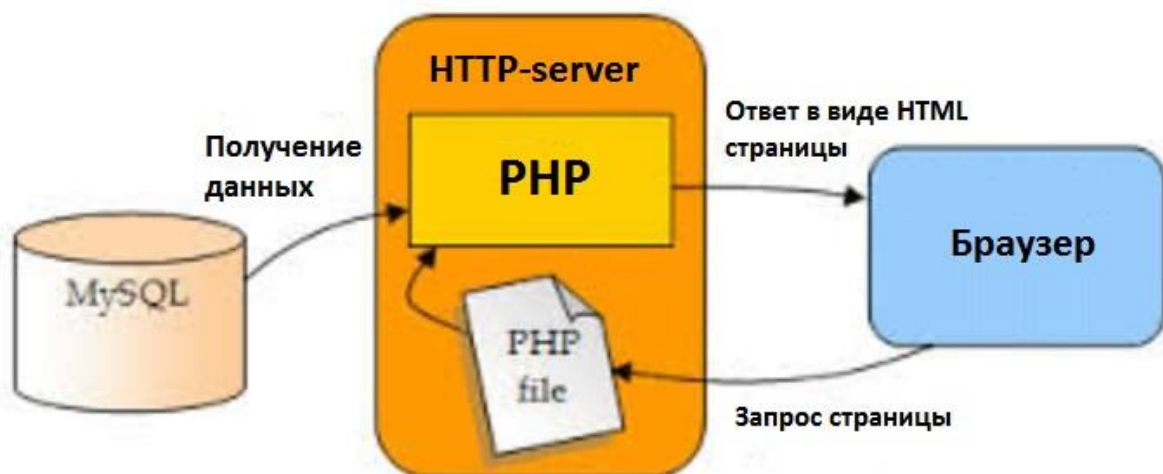


Рис. 15. Схема работы веб-сервера

В качестве языка программирования для написания back-end части web-приложения информационной системы был выбран PHP.

**PHP (препроцессор гипертекста)**, первоначально Personal Home Page Tools — «Инструменты для создания персональных веб-страниц»; произносится пи-эйч-пи — скриптовый язык программирования общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков программирования, применяющихся для создания динамических веб-сайтов [7,13]. Использование PHP осуществляется под PHP License, являющейся open source лицензией, призванной стимулировать широкое распространение исходного кода. Лицензия позволяет распространение материалов в исходной или бинарной форме с изменениями или без таковых при соблюдении следующих условий [17]:

1. включение текста лицензии PHP,
2. слово PHP не может быть включено в название продукта без письменного подтверждения от PHP Group, следующее уведомление должно быть включено в любую форму распространяемых материалов: "This product includes PHP software, freely available from <<http://www.php.net/software/>>".

Если для программного кода на PHP используется лицензия PHP License, то руководство PHP распространяется по лицензии Creative Commons Attribution 3.0 License. Кроме того, некоторые файлы могут распространяться под другими лицензиями. Для распространения таких файлов необходимо соблюдать условия этих лицензий.

Традиционно PHP-код выполняется в php-файлах — это обычные текстовые файлы, имеющие расширение php, редактировать их можно в любом стандартном текстовом процессоре, которые поставляются с любой операционной системой в качестве свободного и бесплатного к использованию текстового редактора [9,13].

В процессе написания использовалась среда разработки программного обеспечения **Geany** (<https://www.geany.org>) (рис. 16.), распространяемая свободно под лицензией GNU GPL. Geany представляет собой текстовый процессор со специальными функциями для разработчиков, такими как: автоматическая подстановка закрывающих тегов, автозавершение названий функций, классов, переменных, подсветка исходного кода, поддержка большого количества кодировок.

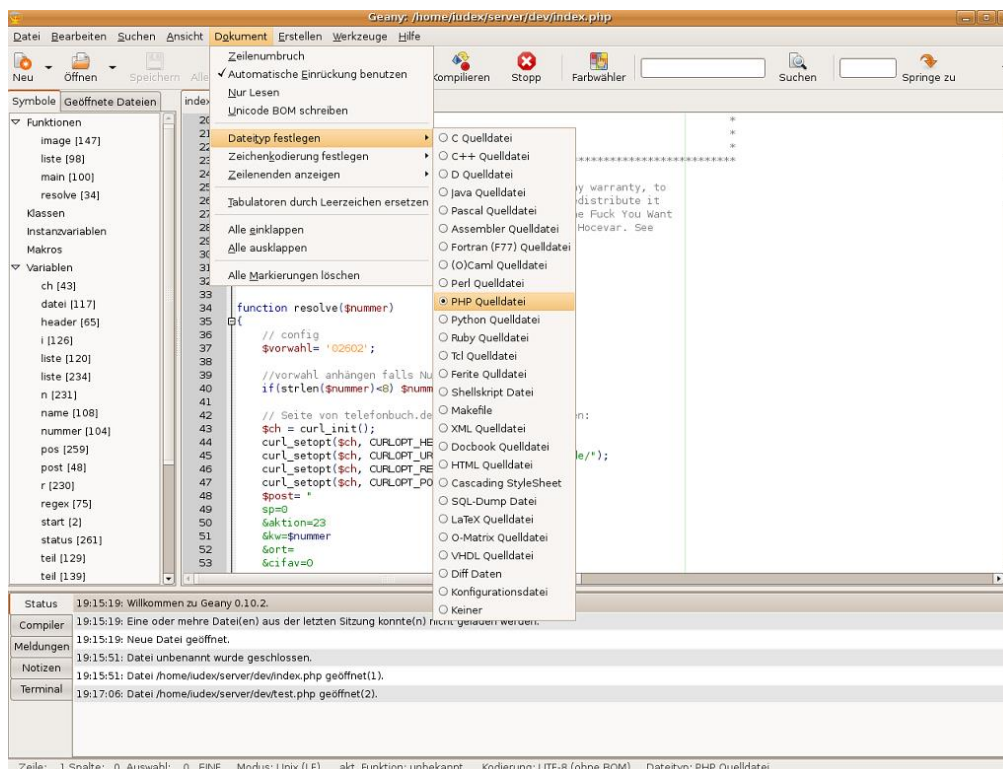


Рис. 16. Текстовый процессор Geany

Написанный PHP код был разделён на различные файлы, в каждом из которых он отвечает за различные задачи. Такое разделение упрощает написание и редактирование проекта, а также даёт возможность подключать только необходимый код для решения текущей задачи пользователя [39].

Файловая структура веб-приложения:

```
./assets/  
    ./ckeditor/  
    ./fancybox/  
    ./favicon/  
    ./fontAwesome/  
    ./base.css  
    ./base.js  
    ./remPolyfill.js  
    ./funcs.js  
    ./normalize.css  
./components/  
    /pages/  
    /safemysql/  
    ./authorization.php  
    ./config.php  
    ./functions.php  
    ./lang.php  
    ./valute_rate.php  
./images/  
./templates/  
    ./404.html  
    ./about.html  
    ./all.html  
    ./authorization.html  
    ./client_edit.html
```

./clients.html  
./goods.html  
./goods\_edit.html  
./logs.html  
./orders.html  
./providers.html  
./provider\_edit.html  
./order\_edit.html  
./passchange.html  
./group\_edit.html  
./main\_right0.html  
./main\_right1.html  
./main\_right2.html  
./ya\_market.html  
./uploads/  
./htaccess  
./COPYING  
./README  
./index.php  
./ya\_market.php

- Директория «assets» содержит вспомогательные файлы, такие как JavaScript скрипты и JavaScript плагины, а также CSS таблицы каскадных стилей (приложение 1) и файлы шрифтов.
- Директория «components» содержит основной PHP-код веб-приложения.
- Директория «images» содержит изображения.
- Директория «templates» содержит HTML шаблоны страниц.
- Директория «uploads» хранит в себе загруженные пользовательские файлы.



- Файлы «COPYING» и «README» являются обязательными файлами для лицензии информационной системы.
- Файл «.htaccess» преобразовывает человеко-удобный HTTP адрес в GET-запросы, которые в свою очередь обрабатывает index.php.
- Файл «valute\_rate.php» производит обновление курса валют, запрашивая данную информацию на сайте ЦБРФ и затем обновляя её в базе данных. Данный файл запускается автоматически в планировщике.
- Файл «index.php» самый важный файл. Он подключает все остальные компоненты веб-приложения в зависимости от того, какую страницу запросил пользователь (приложение 1).
- Файл «ya\_market.php» выгрузка для интеграции ИС с рекламной площадкой «Яндекс.Маркет». Содержит информацию об организации и список товаров.

Используемые информационной системой компоненты, разработанные сторонними разработчиками:

- **SafeMySQL 1.0.0** - PHP класс для безопасного и удобного обращения с MySQL-запросами. Лицензия: Apache License 2.0. (<http://phpfaq.ru/safemysql>)
- **PHPMailer 6.0.1** - PHP класс для создания и отправки писем по электронной почте. Лицензия: GNU LGPLv2.1. (<https://github.com/PHPMailer/PHPMailer>)
- **jQuery 1.12.4** - JavaScript библиотека, для лёгкого доступа к любому элементу DOM, обращений к атрибутам и содержимому элементов DOM, манипуляциям с ними. Лицензия: MIT License. (<https://jquery.com>)
- **fontAwesome 4.7.0** - шрифт/набор иконок с CSS инструментарием. Лицензии: SIL OFL 1.1, MIT License, CC BY 3.0. (<http://fontawesome.io/>)
- **Fancybox 3** - jQuery плагин для отображения в модальных окнах изображений, видео и другого контента. Лицензия: GNU GPLv3. (<http://fancybox.net/>)

- **css3-mediaqueries** - JavaScript скрипт для поддержки CSS Media Queries в устаревших браузерах. Лицензия: MIT License. (<https://github.com/livingston/css3-mediaqueries-js>)
- **HTML5 Shiv 3.7.3** - JavaScript скрипт для поддержки HTML5 тегов в устаревших браузерах. Лицензии: MIT и GPL2. (<https://github.com/aFarkas/html5shiv>)
- **remPolyfill 1.3.2** - Javascript скрипт для поддержки единиц измерения rem в CSS в устаревших браузерах. Лицензия: MIT License. (<https://github.com/chuckcarpenter/REM-unit-polyfill>)
- **CKEditor 5 1.0.0-alpha.2** - Набор компонентов на JavaScript, позволяющий создать удобное решение для редактирования текста. Лицензии: GPL, LGPL, MPL. (<https://github.com/ckeditor/ckeditor5>)
- **Normalize.css 7.0.0** - это небольшой CSS-файл, подключение которого обеспечивает лучшую согласованность между браузерами при стандартном стиле HTML-элементов. Это современная, готовая HTML5, альтернатива традиционному сбросу CSS. Лицензия: MIT. (<https://github.com/necolas/normalize.css>)
- **Google Charts** - это инструмент для построения различных диаграмм в веб-приложениях, предоставляемый посредством API Google. Бесплатное использование согласно условиям, Google APIs Terms of Service. (<https://developers.google.com/chart/>)
- **Jquery UI 1.12.1** - JavaScript набор взаимодействий, эффектов, виджетов и тем, построенных на основе jQuery JavaScript Library. Лицензия: GNU GPLv3. (<http://jqueryui.com/>)

**В целом PHP-код выполняет следующие задачи:**

- Подключается к базе данных
- Осуществляет запросы к базе данных
- Обработывает данные отправленные пользователем
- Обработывает данные полученные из БД
- Загружает соответствующий странице HTML-шаблон

- Вставляет обработанные данные в HTML-шаблон
- Отправляет готовую страницу в веб-обозреватель пользователям

Следующий шаг по созданию информационной системы — это разработка графического интерфейса для ввода, вывода, объединения данных в информацию и других полезных операций в бизнес-деятельности торговой организации.

## 2.2. Разработка клиентской части информационной системы

Front-end частью проекта являются отдаваемые клиенту back-end частью HTML страницы, которые содержат HTML, CSS, JavaScript, изображения и шрифты.

**HTML** — язык гипертекстовой разметки — стандартный язык разметки документов в глобальной сети. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами и отображается в виде документа в удобной для человека форме [32,37].

Язык HTML является приложением SGML (стандартного обобщённого языка разметки) и соответствует международному стандарту ISO 8879.

**CSS** — каскадные таблицы стилей — формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к любым XML-документам, например, к SVG или XUL [35].

**JavaScript** — мультипарадигменный (объектно-ориентированный, императивный, функциональный) язык программирования. В браузерах используется в качестве языка сценариев, придавая странице черту интерактивности. Название языка является товарным знаком Oracle

Corporation, однако сам язык никому не принадлежит (<https://developer.mozilla.org/en-US/docs/Web/JavaScript>) [31,33].

Когда back-end составляющая веб-приложения присылает пользователю html/css (также возможен javascript и иной код) вместе с содержимым, которое запросил человек, браузер пользователя должен вывести на экран готовую страницу. Для этого он преобразует HTML-разметку в объектную модель документа (DOM), а CSS-разметку - в объектную модель таблицы стилей (CSSOM), т.е. создаёт DOM (рис. 17.) и CSSOM (рис. 18.) модели.

```
<html>
  <head>
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <link href="style.css" rel="stylesheet">
    <title>Critical Path</title>
  </head>
  <body>
    <p>Hello <span>web performance</span> students!</p>
    <div></div>
  </body>
</html>
```

Рис. 17. Пример модели DOM

### **Основные этапы формирования DOM:**

1. Преобразование. Браузер преобразует байты из HTML-файла в символы, основываясь на приведенной в файле кодировке (например, UTF-8).
2. Разметка. На основании стандарта W3C HTML5 браузер выделяет среди символов теги в угловых скобках, такие как, и другие. У каждого тега есть свое значение и свой набор правил.
3. Создание объектов. С помощью HTML-тегов браузер выделяет в документе объекты с определенными свойствами.
4. Формирование DOM. Объекты образуют древовидную структуру, повторяющую иерархию HTML-файла, в котором одни теги помещаются в другие. Так, объект p помещается под body, а объект body, в свою очередь, под html, и так далее.

В результате образуется объектная модель документа (DOM), с помощью которой браузер продолжает обрабатывать страницу.

Все эти действия (преобразование байтов в символы, определение разметки, создание объектов и формирование DOM) браузер должен выполнять каждый раз при обработке HTML-разметки. Этот процесс занимает некоторое время, особенно при обработке большого количества тегов которыми реализуется разметка гипертекста.

Чтобы обработать таблицу каскадных стилей CSS, браузер должен выполнить те же самые действия, что и с HTML-документом: байты из CSS-файла преобразуются в символы, символы - в теги, а теги - в объекты, которые образуют модель CSSOM.

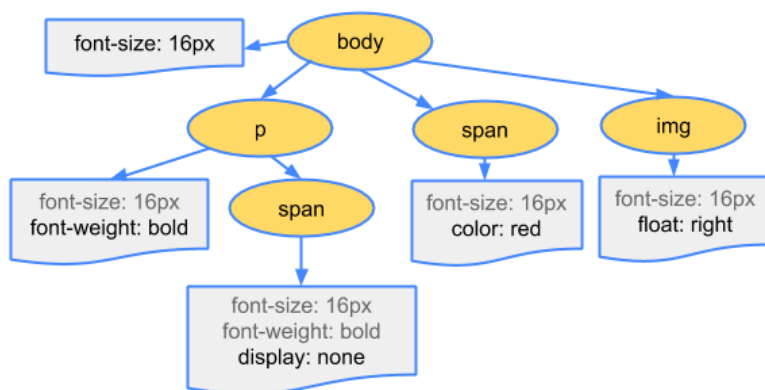


Рис. 18. Пример CSSOM модели

В PHP-коде информационной системы торговой организации написана специальная функция `data2template`. При использовании этой функции код посылает ей данные, которые мы запросили из БД. Затем функция запрашивает тот шаблон, который соответствует странице на которой находится пользователь. Для каждой из страниц уже создан соответствующий шаблон:

`goods_edit.html` – страница редактирования товара;

`provider_edit.html` – страница редактирования поставщика;

`orders.html` – страница заказов;

и т.д.

Такие же шаблоны есть и для иных страниц, отображаемых пользователю информационной системы.

Внутри шаблонов содержится HTML определённой страницы и метка, говорящая PHP-коду (функции data2template) куда вставить данные в этом HTML шаблоне. Меток может быть несколько.

Пример шаблона для страницы «Товары бренда Curaprox»:

```
<div class="mainbox">
    <h2>:                                     <!--[brand]--></h2>
    <table><thead><td>Наименование</td><td>Бренд</td><td>Цена</td></t
head><tbody><!--[table]--></tbody></table>
</div>
```

В приведённом примере можно увидеть 2 метки для данных: <!--[brand]--> и <!--[table]-->. Вместо первой функция вставит название бренда, а вместо второй сформированную в PHP-коде таблицу с товаров этого бренда.

Приведённые выше шаблоны не являются полноценными HTML страницами, а содержат лишь основу контента веб-приложения. В них отсутствуют такие обязательные теги HTML как body, head, также в них отсутствует основной дизайн приложения и CSS.

Так как основные элементы дизайна веб-приложения являются статическими (шапка, подвал, фон), то для удобства они были выделены в отдельный шаблон all.html. Таким образом, при необходимости изменить основные элементы разметки веб-приложения отпадает надобность менять их в шаблонах каждой страницы, изменения нужны только в вышеупомянутом файле.

Файл all.html используется при загрузке любой страницы приложения и содержит метки: заголовок страницы, контент страницы, дополнительный код между <head></head> тегами.

**Логика использования шаблонов выглядит следующим образом:**

1. Приложение получает запрос на выбранную пользователем страницу;
2. Запрашиваются данные соответственно выбранной страницы из БД;
3. Приложение обрабатывает данные для их дальнейшего отображения пользователю (к примеру, составляет таблицу);

4. Загружается шаблон, соответствующий странице;
5. Согласно меткам, приложение располагает обработанные данные в шаблоне;
6. Загружается основной шаблон all.html, согласно его меткам, в него вставляются заголовок страницы и уже обработанный шаблон страницы;
7. Полученный шаблон отправляется веб-обозревателю пользователя.

Шаблон all.html является основой и большей частью веб-приложения, в нём подключаются скрипты JavaScript и таблицы CSS. Начинается страница с указания (веб-обозревателю) типа текущего документа DTD (document type definition, описание типа документа): `<!DOCTYPE html>` - обозначает новейшую, 5 версию HTML.

В интернете долгое время использовались стандарты HTML 4.01, XHTML 1.0 и XHTML 1.1. Веб-страницы на практике верстались с использованием смешанных особенностей, представленных различными спецификациями, включая спецификации программных продуктов, например, веб-браузеров, а также сложившихся общеупотребительных приёмов. HTML5 был создан как единый язык разметки, который мог бы сочетать синтаксические нормы HTML и XHTML. Он расширяет, улучшает и рационализирует разметку документов, а также добавляет единый API для сложных веб-приложений.

**После объявления типа документа идут основные элементы для html-страниц (рис. 19):**

- html - элемент верхнего уровня в документе, для HTML;
- head - предназначен для хранения других элементов, цель которых — помочь браузеру в работе с данными. Между открывающим и закрывающим тегами head указана кодировка, язык, подключаемые файлы (CSS, JavaScript и шрифты).
- body - предназначен для хранения содержания веб-страницы (контента), отображаемого в окне браузера. Т.е. между открывающим и

закрывающим тегами данного элемента находится всё то, что видит пользователь: «шапка» сайта, заголовки, контент страницы и «подвал» сайта.

```
<!DOCTYPE html>
<html lang="ru">
<head> <!--голова сайта-->
    <meta charset="utf-8">
    <meta name="description" content="Создание сайтов на HTML5">
    <meta name="keywords" content="HTML5, CSS3, JavaScript">
    <title>Разработка сайтов с помощью HTML5</title>
    <link rel="stylesheet" href="style.css">
</head>
<body> <!--тело сайта-->
</body>
</html>
```

Рис. 19. Пример структуры HTML-документа

В приложении используется кодировка UTF-8, она указана как в «head» HTML, так и в начале основного PHP-файла. Кроме того, базы данных также хранят информацию в данной кодировке.

UTF-8 (от англ. Unicode Transformation Format, 8-bit — «формат преобразования Юникода, 8-битный») — одна из общепринятых и стандартизированных кодировок текста, которая позволяет хранить символы Юникода, используя переменное количество байт (от 1 до 6).

Стандарт UTF-8 официально закреплён в документах RFC 3629 и ISO/IEC 10646 Annex D. Кодировка нашла широкое применение в UNIX-подобных операционных системах и веб-пространстве [13]. Сам же формат UTF-8 был изобретён 2 сентября 1992 года Кеном Томпсоном и Робом Пайком и реализован в Plan 9[2]. В качестве BOM использует последовательность байт EF16, BB16, BF16 (что у неё самой является трёхбайтовой реализацией символа FEFF16).

Если HTML указывает где отображаются элементы веб-страницы, то CSS-правила задают как именно. Сами правила также содержатся в основном шаблоне и прописаны в специальном HTML-теге <style type="text/css">.



Перед тем, как описывать стили HTML-элементов в CSS был добавлен `normalize.css` — это набор стилей, который обеспечивает для HTML-элементов лучшую кроссбраузерность в стилях по умолчанию. Проект является продуктом сотен часов глубокого исследования различий между изначальными стилями браузера. Это исследование провели Николас Галахер, `@necolas` и Джонатан Нил, `@jon_neal`. Когда элемент имеет различные стили по умолчанию в разных браузерах, `normalize.css` там, где это возможно, стремится сделать эти стили совместимыми и соответствующими современным стандартам.

### Цели `normalize.css`:

- сохранять полезные настройки браузера, а не стирать их;
- нормализовать стили для широкого круга HTML-элементов;
- корректировать ошибки и основные несоответствия браузера;
- совершенствовать юзабилити незаметными улучшениями;
- объяснять код, используя комментарии и детальную документацию.

`Normalize.css` свободно распространяется под лицензией MIT.

С помощью CSS в приложении заданы свойства различных элементов дизайна, к примеру, размер, цвет, семейство, вариант написания шрифта (рис. 20.) как для всей страницы, так и отдельно для некоторых элементов, например, заголовков. Заданы общий фон страницы и фон контента, обрамление таблиц и отступы в их ячейках.

```
body {  
    background-color: #d0e4fe;  
}  
  
h1 {  
    color: orange;  
    text-align: center;  
}  
  
p {  
    font-family: "Times New Roman";  
    font-size: 20px;  
}
```

Рис. 20. Пример стилей CSS для HTML-элементов

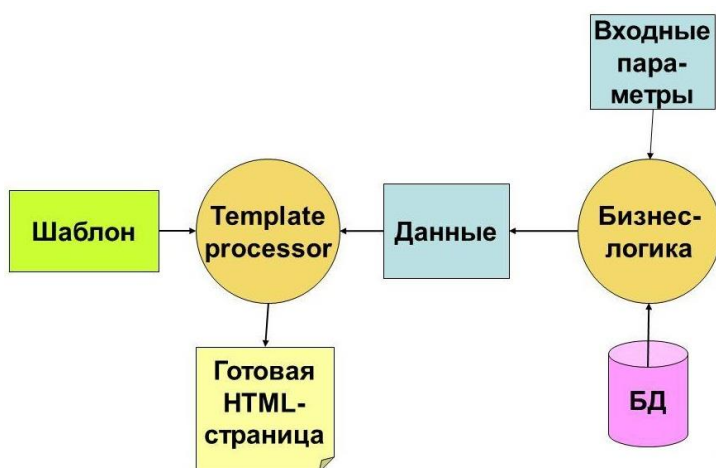


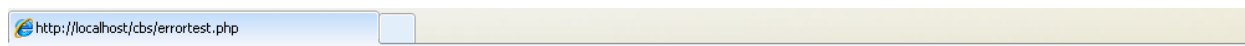
Рис. 21. Диаграмма работы веб-приложения

Разработав back-end и front-end части веб-приложения информационной системы - получен необходимый для функционирования информационной системы функционал. Работая как клиент-серверное приложение, система избавляет конечного пользователя в необходимости установки дополнительного программного обеспечения, т.к. основные процессы (рис. 21.) веб-приложения происходят на стороне сервера.

### 2.3. Тестирование и обзор информационной системы

Особенности тестирования, ориентированного на клиент-серверное взаимодействие веб-приложения, предполагают наличие навыков находить дефекты как на стороне клиента (веб-браузер), так и на стороне сервера. Важно учесть дефекты, возникающие при взаимодействии сервера и клиента, такие дефекты зачастую связаны с проблемами в бизнес-логике (функциональные дефекты) информационной системе. Функциональное тестирование направлено на то, чтобы каждая функция веб-сайта работала в соответствии с требованиями спецификации. Тестирование функциональности веб-сайта показывает «Что делает система». Именно поэтому важно изначально проанализировать аспекты разрабатываемой информационной системы и провести проектирование, такой подход изначально позволит избежать данных ошибок.

Для нахождения ошибок в коде разработанного веб-приложения были использованы стандартные средства и методики. Для PHP-кода – это вывод ошибок при исполнении скрипта. Такое тестирование производилось в процессе написания кода и проверки его функционирования. При значимых, критических ошибках PHP-код не исполняется и выводит соответствующее сообщение, указав на проблемный участок кода (рис. 21.).



**Parse error:** syntax error, unexpected T\_IF in C:\Inetpub\wwwroot\cbs\errorrest.php on line 4  
PHP Parse error: syntax error, unexpected T\_IF in C:\Inetpub\wwwroot\cbs\errorrest.php on line 4

Рис. 21. Вывод PHP-ошибки в браузере

В PHP имеется широкий функционал по настройке вывода и обработки таких ошибок – их можно выводить в клиентский браузер, специальный текстовый файл, записывать в базу данных, никуда не выводить. PHP предоставляет функционал для создания собственных обработок ошибок, благодаря которым, их можно дополнить ошибками с собственной классификацией (к примеру при ошибках в бизнес-логике). Стандартно ошибки в PHP классифицируются так:

- фатальные (неустраняемые, при возникновении таких скрипт прекращает работу);
- не фатальные (устраняемые, скрипт не прекращает работу);
- смешанные (фатальные, но только, если не обработаны функцией, определенной пользователем);
- предупреждения (не являются ошибками, однако, указывают на недостатки в исполняемом коде).

Существует и реже используемый способ проверки PHP кода на наличие ошибок – это стороннее программное обеспечение, проверяющее загруженный в них код или онлайн сервисы, как правило они проверяют только синтаксис, не исполняя его. Примером может служить сервис на сайте Piliapp.com (<https://ru.piliapp.com/php-syntax-check/>).

Тестирование ошибок на стороне клиента (в веб-браузере) во многом осуществлялось благодаря предоставленному в браузерах инструментарию для веб-разработчиков (рис. 22.).

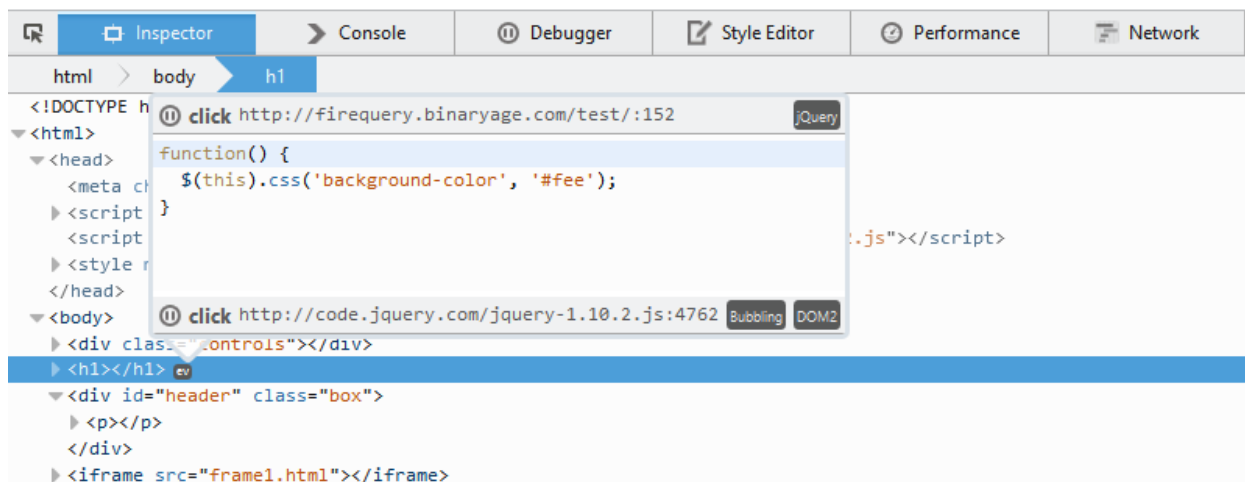


Рис. 22. Инструменты веб-разработчиков в Mozilla Firefox

**Инструменты для веб-разработчиков** – это комплекс средств, представляющий из себя специальную панель и включающий такие средства как[25]:

- Инспектор – даёт возможность редактировать DOM-модель в реальном времени (все изменения сразу отображаются на странице);
- Консоль – незаменимый инструмент, показывающий ошибки в выполнении JavaScript, CSS, ошибки при загрузке файлов и т.д.;
- Запросы (сеть) – показывает все http запросы, их заголовки, ответы, содержание. Отобразит соответствующую информацию, если к примеру, не удалось загрузить изображение с сервера;
- Хранилище – показывает и позволяет редактировать данные содержащиеся в различных хранилищах клиента, таких как Cookie, Indexed DB, сессионное хранилище, хранилище кэша.
- Стили – похож на инспектора, но ориентирован на CSS;
- Память – показывает на выполнение и загрузку чего тратиться соответствующий ресурс клиента;

- Профайлер – мощный инструмент для глубокого анализа процессов происходящих при загрузке страницы, построении DOM, применении стилей и многого другого;
- Адаптивный дизайн – помогает разрабатывать и проверять страницу на правильное отображение в экранах разного размера, что особенно актуально в нынешнее время, когда сайтами в равной степени пользуются как с мобильных устройств, так и с десктопных.

Также в зависимости от браузера и его версии присутствуют и другие инструменты, такие как линейка, скриншот страницы, пипетка, редактор JavaScript и другие.

Написание выполняющегося кода не всегда является залогом его эффективного функционирования и получения одинакового результата в различных браузерах. Такая проблема совместимости во многом исключается при написании кода с соблюдением соответствующих стандартов и исключается полностью при тестировании дополнительными проверками.

Для этого W3C (Консорциум Всемирной паутины) создал автоматизированные тесты (рис. 23.) совместимости для CSS, HTML и JavaScript. Данные тесты производятся как на отдельных участках кода, так и на готовых веб-приложениях доступных в глобальной сети. По результатам проверки выводятся рекомендации по исправлению. Такие проверки дополняют многие IT-гиганты (такие как Google и Яндекс) заинтересованные в росте качества интернет ресурсов [11]. Популярнейшим тестом у корпорации Google [1,11], является тест на скорость (рис. 24.) загрузки веб-страниц и наличие свойства адаптивности у веб-интерфейса для отображения на различных устройствах. Яндекс в 2017 году встроил комплекс таких и других проверок для пользователей сервиса «Яндекс Вебмастер» [24,29,36] (рис. 25.).

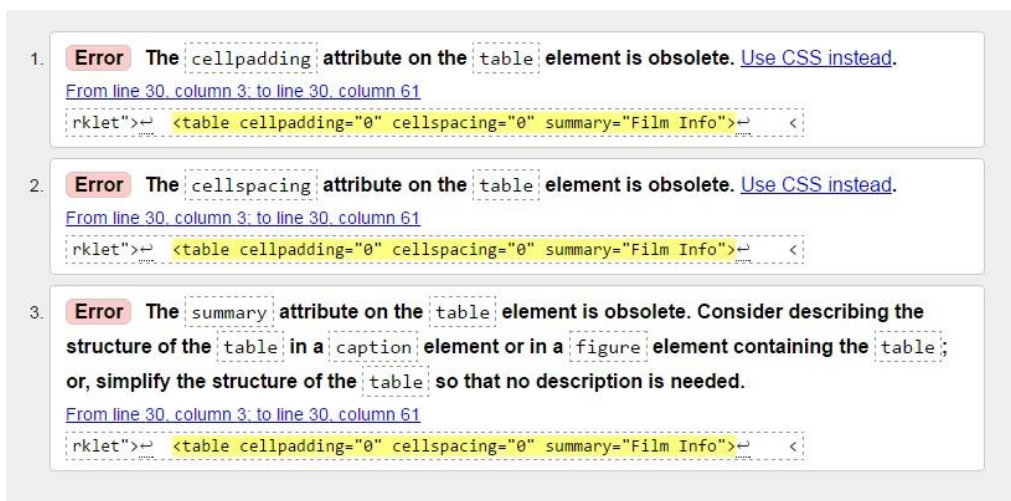


Рис. 23. Ошибки в HTML показанные валидатором кода от W3C

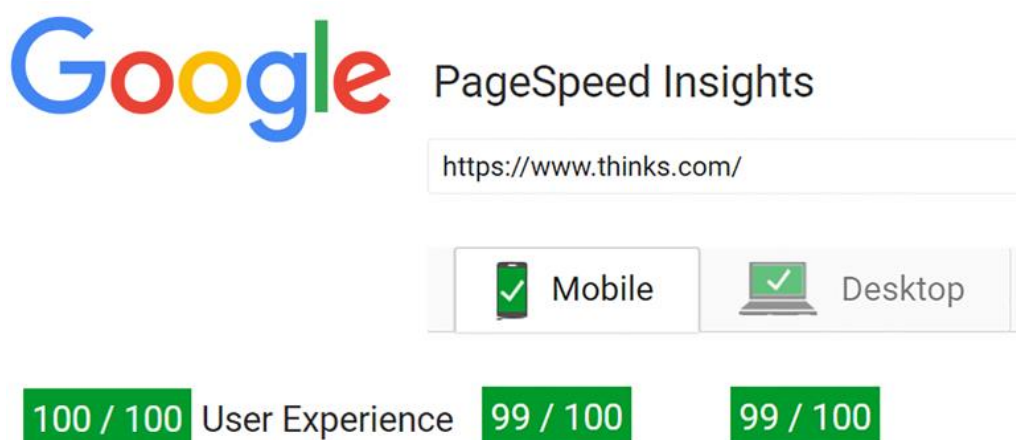


Рис. 24. Google PageSpeed Insights – проверка на скорость загрузки сайта

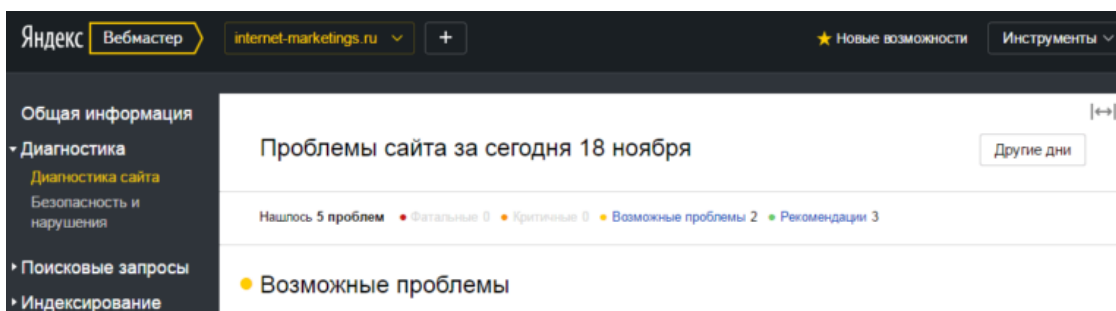


Рис. 25. Диагностика ошибок в Яндекс.Вебмастере

Кроме стандартных и автоматизированных средств и методик нахождения дефектов, каждый веб-разработчик с появлением опыта начинает составлять свою стратегию и свои «чек листы» для проверки информационных систем и их составляющих. Такие проверки могут быть узконаправленными и составляться непосредственно для данной системы во

время проектирования или иметь список для проверки частых ошибок в любой информационной системе.

Чек-лист для тестирования веб-интерфейса пользователя:

- Тестирование ссылок
  - Исходящие ссылки
  - Корректность внутренних ссылок
  - Отсутствие ссылок, ведущих к одной странице
  - Есть ли страницы, на которые не указаны ссылки
  - Отсутствие неработающих ссылок
- Тестирование форм для всех страниц
  - Действительность входных данных
  - Допустимые значения для поля данных
  - Недопустимые входные значения для поля данных
  - Параметры форм, в которых возможно удаление или любая другая модификация данных.
- Тестирование cookies
  - Проверьте сайт с отключенными cookies
  - Проверьте сайт с включенными cookies
  - Убедитесь, что файлы cookies зашифрованы перед записью на компьютер пользователя
  - Проверьте аспекты безопасности при удалении файлов cookies.
  - Если cookies имеют продолжительность действия, то следует проверить, активны ли они в указанный период времени.
- Убедитесь, что сайт доступен для поисковых машин.

Во многом тестирование упрощает унифицированное написание кода и повторное использование компонентов. В среде веб-интерфейсов в этом способствует разработка веб-приложения по методологии БЭМ, а при написании PHP-кода стандарты PSR. Немаловажным является подготовка к созданию базы данных, которая предполагает её проектирование путём составления инфологической и даталогической моделей.

Обследуя область тестирования веб-приложения нельзя не упомянуть маркетинговые исследования. Такие исследования должны проводиться специалистами компетентными в сфере маркетинга, однако, по ранее описанным методологиям веб-разработки, каждый участник команды (или специалист единолично разрабатывающий ИС) должен иметь понимание и долю компетенции на каждой ступени разработки и проектирования информационной системы. Маркетинговые исследования в области разработки информационных систем являются формой бизнес-исследования, сфокусированного на понимании поведения, желаний и предпочтений потребителей конечного продукта, а также предпочтений, диктуемых рынком экономики. Такие исследования являются важной и неотъемлемой частью как проектирования, так и тестирования ИС в экономическом секторе.

Наиболее универсальным и распространённым методом в маркетинговых исследованиях является А/В тестирование. Цель данного тестирования – это улучшение целевого показателя. Достигается это путём сравнения контрольной группы с тестовыми группами, в которых один или несколько показателей были изменены. Такой метод часто используется в веб-приложениях для улучшения показателей дизайна, из типичных примеров можно выделить: влияние цветовой схемы, расположение и размеры элементов веб-интерфейса сайта с целью увеличить важный целевой показатель - конверсия. В таком примере А/В тестирование показывает разным пользователям разный вариант исполнения дизайна, а по накоплению необходимого числа показов, происходит сравнение числовых показателей достижения цели, из чего выбирается наиболее подходящий вариант дизайна.

Использование в комплексе вышеперечисленных методов и инструментов для тестирования ошибок в коде, логике и процессах взаимодействия информационных систем, основанных на веб-приложениях, позволило не только исключить возникновение ошибок, но и повысить рабочие характеристики и важные показатели на более качественный уровень.

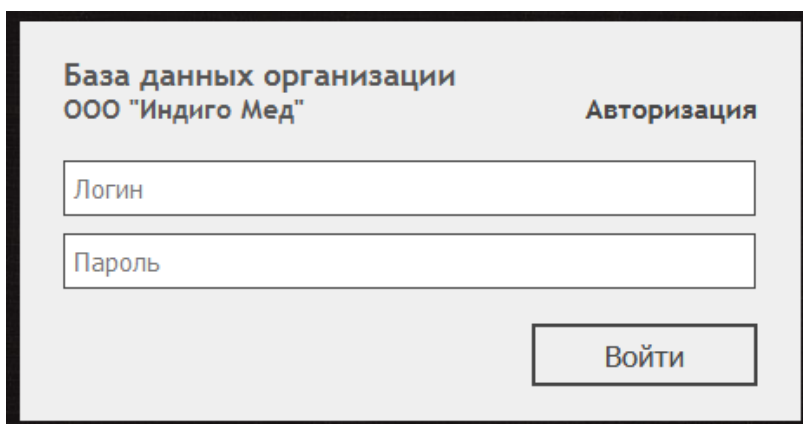


Обзор информационной системы необходим для выявления полезных свойств системы, решающих задачи конечного потребителя. Данный обзор затрагивает функциональную составляющую, с которой пользователь работает непосредственно, а также общую информацию об информационной системе.

Разработанная информационная система для торговой организации является клиент-серверной реализацией. Серверная часть имеет специальное программное и аппаратное обеспечение, отвечающее за хранение, обработку и доступ к данным. Данное программное обеспечение является свободно распространяемым, активно обновляемым разработчиками и популярным в использовании веб-разработок, доступно к использованию в различных операционных системах, распространяемых как на проприетарных лицензиях, так и на свободных. Клиентская часть информационной системы предполагает любую (уже имеющуюся) операционную систему с наличием графического интерфейса на устройстве клиента, наличие интернет соединения и веб обозревателя с актуальной версией. Рекомендованным браузером для использования системы, является свободно распространяемый браузер Firefox последней версии, от компании Mozilla.

Обзор разделов веб-приложения и их взаимодействия при работе пользователя с информационной системой:

#### **«Страница авторизации»**



База данных организации  
ООО "Индиго Мед"

Авторизация

Логин

Пароль

Войти

Рис. 26. Блок авторизации

Первая страница, которую увидит пользователь информационной системы, находясь в неавторизованном состоянии. В целях безопасности доступ к работе с веб-приложением осуществляется по логину и паролю. Возможность регистрации отсутствует, т.к. информационная система нацелена на работу с ограниченным кругом лиц, имеющим доступ к информации. Первично доступ для руководителя организации предоставляет разработчик или настройщик информационной системы, доступ иным сотрудникам может предоставить представитель организации на странице «Администрирование». При попытке войти под парой логин и пароль, система сравнивает их на соответствие в базе данных информационной системы. При успешном входе клиент (браузер) пользователя получает доступ к функционалу информационной системы, при неудачном сравнении пары логин и пароль, пользователь получает отказ и может повторить попытку.

### «Главная панель»

Торговая база ООО "Индиго Мед"	Терехов Григорий (Программист)	Продажи: 2	Задачи: 1	Администрирование	Сменить пароль	Выход
Главная	Заказы	Клиенты	Поставщики	Товары	Задачи	Статистика

Рис. 27. Главная панель

Важный элемент, присутствующий на каждой странице веб-приложения авторизованного пользователя. Осуществляет функцию навигации между основными страницами веб-приложения, содержит название организации, под которую настроена торговая база данных, Ф.И.О. и должность авторизованного пользователя, показывает кол-во заказов «в работе» данного менеджера и аналогично кол-во задач.

### «Сменить пароль»

Торговая база ООО "Индиго Мед"	Терехов Григорий (Программист)	Продажи: 2	Задачи: 1	Администрирование	Сменить пароль	Выход
Главная	Заказы	Клиенты	Поставщики	Товары	Задачи	Статистика

Смена пароля

Введите новый пароль

Сохранить

Рис. 28. Смена пароля

Данная страница позволяет поменять пароль пользователя для авторизации в ИС.

## «Информация о системе»

Торговая база  
ООО "Индиго Мед"

Терехов Григорий (Программист)

Продажи: 2    Задачи: 1

Администрирование    Сменить пароль    Выход

Главная

Заказы

Клиенты

Поставщики

Товары

Задачи

Статистика

Информация о системе

**Система управления торговой базой v.1.1 от 1 ноября 2017**

Система имеет единую базу данных для хранения, обработки и добавления данных о товарах, заказах, контрагентах. Имеется широкий функционал по автоматизации бизнес процессов коммерческой организации в работе с данными базы, мониторингом работы сотрудников, формирования отчетов и построения задач. Имеются инструменты интеграции со сторонними сервисами, средства связи между сотрудниками и сотрудников с клиентами. Информационная система не требует дополнительного и специфического программного обеспечения на клиентском оборудовании, имеет адаптивность дизайна под различные клиентские устройства. Система предусматривает возможность разграничения прав доступа к данным базы разным сотрудникам предприятия.

**Минимальные требования к аппаратному и программному обеспечению**

Для сервера с 20 подключёнными пользователями:  
 Процессор - Intel Core i3-2xxx, либо аналогичный  
 Оперативная память - 1 гб  
 Жесткий диск - 1 гб  
 Интернет соединение - 8 мбит  
 Выделенный IP-адрес  
 PHP версии >= 5.6  
 MySQL server версии >= 5.7  
 MySQL client версии >= 5.7  
 Apache >= 2.4

Рис. 29. Информация о системе

Страница, на которую можно пройти по ссылке внизу каждой авторизованной страницы ИС. Отражает текущую версию информационной системы, её описание, минимальные требования, перечень изменений в последнем обновлении и планы разработчика на следующее, контакты и наименование разработчика, информацию о сторонних компонентах веб-приложения и информация о лицензии данной ИС.

## «Главная страница»

Заказы в работе

№	Дата	Клиент	Статус	Отправлен	Сумма
2	22 октября 00:00	Василий Геннадьевич Иностранов	Завершён	Нет	5343 р.
1	5 ноября 00:00	Василий Геннадьевич Иностранов	Завершён	Нет	4123 р.

Последние действия пользователей

Дата	Пользователь	Действие
14 декабря в 14:44	Терехов Григорий	Клиент отредактирован
11 декабря в 14:40	Терехов Григорий	Группа отредактирована
11 декабря в 14:40	Терехов Григорий	Группа отредактирована
11 декабря в 14:40	Терехов Григорий	Группа отредактирована
11 декабря в 14:38	Терехов Григорий	Группа отредактирована
11 декабря в 14:38	Терехов Григорий	Группа отредактирована
11 декабря в 14:08	Терехов Григорий	Изменения пользователя сохранены
11 декабря в 14:08	Терехов Григорий	Изменения пользователя сохранены
11 декабря в 13:53	Терехов Григорий	Изменения пользователя сохранены
11 декабря в 13:53	Терехов Григорий	Изменения пользователя сохранены

Сообщения

Терехов Григорий (1 декабря в 05:13)

Мой чат :)

01 декабря 14:44, Пятница

USD 58.9281    EUR 69.706

Курс доллара и евро за 7 дней

Документы

Реквизиты организации

Свидетельство о регистрации

Свидетельство из налоговой

Прайс-лист (excel)

Сводка

Завершённые заказы

Текущий месяц: 0

Прошлый месяц: 0

Новые клиенты

Текущий месяц: 0

Прошлый месяц: 0

Рис. 30. Главная страница

Является общим обзором текущей деятельности организации и содержит дополнительные полезные материалы для коммерческой организации. Руководители организации могут видеть все продажи (заказы) находящиеся в работе (не имеющие статус завершённых или неотправленных покупателю), торговые менеджеры могут увидеть свои заказы в соответствующем состоянии «в работе». Далее на странице присутствует таблица со списком последних действий пользователей системы, связанных с бизнес-процессами – это помогает лучше координировать работу сотрудников, за ней есть модуль внутреннего чата информационной системы, созданного с той-же целью. Посредством него можно оставить краткую, но важную информацию для всех сотрудников организации. В блоке дополнительной информации присутствует текущие дата и время, ежедневно обновляемые курсы доллара и евро к рублю, а также график их изменения за неделю, важные и актуальные документы организации, которые полезно хранить «под рукой», небольшая месячная сводка по привлечению новых клиентов и продаж.

### «Заказы»

Заказы в работе						Добавить заказ	
№	Дата	Клиент	Статус	Отправлен	Сумма		
2	22 октября 00:00	Василий Геннадьевич Иностранов	Завершён	Нет	5343 р.		
1	5 ноября 00:00	Василий Геннадьевич Иностранов	Завершён	Нет	4123 р.		

Статистика			
Декабрь 2017	1 шт.	43534 р.	
Ноябрь 2017	1 шт.	4123 р.	
Октябрь 2017	2 шт.	62099 р.	

Все заказы						
№	Дата	Клиент	Статус	Отправлен	Сумма	
4	22 октября 00:00	Анатолий Георгиевич Потапов	Завершён	Да	56756 р.	

Рис. 31. Главная страница раздела «Заказы»

Этот раздел позволяет работать с соответствующей названию информацией. На основной странице раздела имеются 2 таблицы: заказы «в работе» и все заказы (отсортированные по убывающей дате поступления). Также на странице есть автоматически формирующаяся статистика

завершённых заказов за последние месяцы. В этом разделе можно добавлять заказы, редактировать и удалять, оставлять по каждому из них заметки, сформировать версию для печати, распечатать товарный чек, автоматически сформировать и отправить клиенту на электронную почту соответствующее статусу его заказа письмо.

### «Клиенты»

The screenshot displays the 'Clients' section of a software interface. It features three main components:

- Новые и активные (New and Active):** A table listing two clients with their names, email addresses, phone numbers, and action icons (edit, delete, favorite).
- Добавить клиента (Add Client):** A button in the top right corner.
- Новые клиенты (New Clients):** A summary showing 5 clients in November 2017 and 1 client in July 2017.
- Отмеченные вами (Marked by You):** A list of three clients marked by the user: Анатолий Георгиевич Потапов, Василий Геннадьевич Иностранов, and ООО "Зонд".
- Все клиенты (All Clients):** A table at the bottom listing all clients, including their addresses and the number of orders placed.

Наименование	Эл.почта	Телефон	
Анатолий Георгиевич Потапов	potap@example.ru	8 922 33232 32	
Василий Геннадьевич Иностранов	inostr@example.ru	8 800 2000 500	

Ноябрь 2017	5
Июль 2017	1

Анатолий Георгиевич Потапов	
Василий Геннадьевич Иностранов	
ООО "Зонд"	

Наименование	Эл.почта	Телефон	Адрес	Заказов	
Анатолий Георгиевич Потапов	potap@example.ru	8 922 33232 32	г. Нижний Тагил	2	
Василий Геннадьевич Иностранов	inostr@example.ru	8 800 2000 500	г. Екатеринбург, улица Пушкина 2	2	

Рис. 32. Главная страница раздела «Клиенты»

Этот раздел также имеет главную страницу, на которой расположены 2 таблицы: новые и активные клиенты, и таблица всех клиентов. Имеется возможность добавления, изменения и удаления клиентов, можно оставить заметку по клиенту и добавить в свой список избранных (отображается в этом же разделе), для более быстрого доступа к ним. Также в редакторе клиентов можно увидеть сколько и на какую сумму клиент делал заказов, в каком состоянии эти заказы, сформировать печатную версию информации о клиенте.

## «Поставщики»

Изменить поставщика — ЗАО "Дентал"

Поставляемые товары ЗАО "Дентал" - 11 шт.

Бренд	Наименование	Артикул
Curaprox	Duo Love - набор зубных щёток UltraSoft	CS 5460/2
Curaprox	Hydrosonic - звуковая щетка	CHS 100
Curaprox	Звуковая щетка Hydrosonic дорожный набор	CHS 100
Curaprox	Зубная щётка "Soft"	CS 1560
Curaprox	Зубная щётка "SuperSoft"	CS 3960
Curaprox	Зубная щётка "UltraSoft"	CS 5460
Curaprox	Prime plus межзубные ёршики 5шт. + держатель UHS451	CPS prime plus
Curaprox	Regular межзубные ёршики для брекет систем 4шт. + держатели	CPS
Curaprox	Soft&Implant межзубные ёршики 100шт. + 4 держателя UHS 409	CPS
Curaprox	Набор "STARTER KIT"	CPS 556
Curaprox	Набор для гигиенического ухода "Дорожный"	CPS 457

Наименование

ЗАО "Дентал"

Эл.почта

dental@example.ru

Телефон

8 (343) 4-31-85

Адрес

г. Екатеринбург, улица Союзов 2

Сохранить

Удалить

Система управления торговой базой v.1.1 от 01.11.2017

Рис. 33. Страница изменения поставщика

Данный раздел показывает какие есть поставщики у организации и сколько товаров они поставляют. Поставщиков можно добавлять, редактировать, удалять. Открыв страницу нужного поставщика — можно увидеть список товаров, которые он поставляет. Товар прикрепляется к поставщику на странице определённого товара, которая будет рассмотрена ниже.

## «Товары»

Группы товаров

Биоматериалы	13		
Индивидуальная гигиена			
Зубные пасты	8		
Зубные щётки	6		
Межзубные ёршики	5		
Ополаскиватели	5		
Инструменты			
Зажимы	4		
Зеркала	11		
Кусачки	9		
Пинцеты	15		
Скальпели	16		
Щипцы			
Щипцы для снятия	7		
Щипцы для установки	4		
Щипцы универсальные	3		
Ортопедические изделия	9		
Товары вне категорий			

Все категории

Все бренды

Добавить группу

Добавить товар

Добавить бренд

Всего товаров: 115

Новые товары:

Ни-Friedy Щипцы универсальные для формирования изгибов на дуге  
 Ни-Friedy Щипцы ортодонтические  
 Ни-Friedy Щипцы Waingart для продвижения дуги  
 Ни-Friedy Щипцы для установки сепарационных колец  
 Ни-Friedy Щипцы для установки металлической лигатуры  
 Ни-Friedy Щипцы Weingart с ультратонким профилем  
 Ни-Friedy Пинцет для постановки брекетов

Рис. 34. Главная страница раздела «Товары»

Один из самых больших и основных разделов веб-приложения. На главной странице в первую очередь показана таблица групп товаров, зайдя в которую можно увидеть товары, которые в ней содержатся. Можно увидеть список последних добавленных товаров, общее количество товаров и специальное меню для перехода к добавлению брендов, групп товаров и самих товаров, к списку брендов. Описывая одним предложением: раздел «Товары» имеет функционал по добавлению, редактированию и удалению брендов, групп товаров и товаров.

### «Статистика»

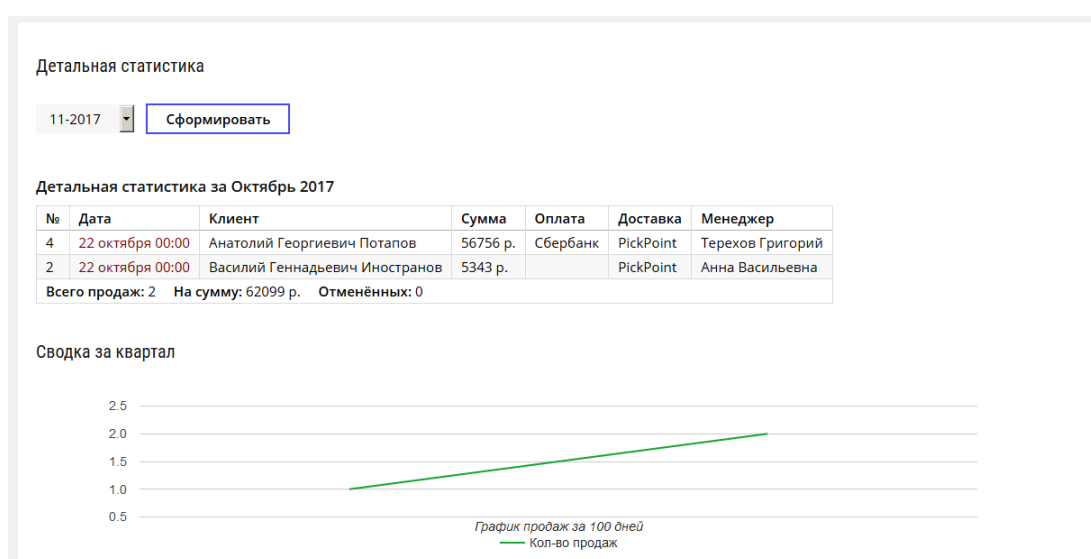


Рис. 35. Раздел статистики

Любая информационная система должна иметь инструменты для анализа бизнес процессов. Отчасти данный функционал реализован в нескольких вышеописанных разделах, однако в этом выступает в более детальном виде, отражая графики продаж и денежного оборота за последние 100 дней, таблицу со сводкой продаж, оборота, среднего чека и другой информацией за квартал. Присутствует возможность сформировать и распечатать детальный отчёт по продажам за необходимый пользователю месяц.

## «Задачи»

Задача — Поиск поставщиков для бренда Cugarprox

Цели и задачи

Необходимо найти поставщиков с ценами на товар на 5% меньше средней цены текущих поставщиков.

План работ

1.Составить средние цены текущих поставщиков

2.Составить потенциальных контрагентов

3.Запросить информацию по ценам

4.Узнать условия дополнительного снижения цен

5.Предоставить варианты сотрудничества

Сообщения

Терехов Григорий (9 декабря в 23:59)

Привет!

Создана: 27 ноября

Срок до: 15 декабря

Приоритет: Высокий

Статус: В работе

Последняя редакция: 10 декабря в 00:16

Руководитель: Анна Васильевна

Участники:

Терехов Григорий

Анна Васильевна

Внести изменения

Рис. 36. Просмотр задачи

Дополнительный инструмент для торговой организации, созданный с целью координации работы сотрудников начиная от мелких задач заканчивая большими проектами. Данное средство позволяет руководителям направлений или руководителю организации создавать задачи для нескольких или всех сотрудников. Каждая задача содержит своё описание, цели, задачи, план работ, сроки, приоритет, ответственного и имеет свой собственный чат для обсуждения процессов достижения цели.

## «Администрирование»

Общие настройки

Название организации:

ООО "Индиго Мед"

Название магазина:

Индиго Мед

ИНН организации:

295720384

ОГРН организации:

2387936723952

Телефон магазина:

8 800 2322 232

Адрес магазина:

Екатеринбург, ул. Торговая 5

Банковские реквизиты:

Банк "Нейва"

Электронная почта:

noeprly@torg-base.ru

Пароль эл. почты (для авто-отправлений):

SMTP сервер эл. почты (для авто-отправлений):

smtp@yandex.ru

Настройки Яндекс.Маркет

Включить интеграцию:

Управление пользователями

Анна Васильевна (anna)

Логин: anna

Пароль:

Права: Менеджер

Имя: Анна Васильевна

Должность: Менеджер

Применить

Терехов Григорий (grig)

Логин: grig

Пароль:

Права: Администрат

Имя: Терехов Григорий

Должность: Программист

Применить

Эльмира Радикова (elya)

Логин: elya

Пароль:

Права: Руководитель

Имя: Эльмира Радикова

Должность: Директор

Применить

Новый пользователь

Логин:

Пароль:

Права: Администрат

Имя:

Должность:

Добавить

Смотреть логи

Рис. 37. Страница Администратирования



Данная страница доступна руководству организации и администратору ИС. Здесь присутствуют основные настройки информационной системы, настройки интеграции со сторонними ИС, функционал по управлению пользователями.

### «Логи»

Логи			
Дата	Пользователь	Действие	
14 декабря 2017 в 14:44	Терехов Григорий	Клиент отредактирован	Все логи
14 декабря 2017 в 14:39	Терехов Григорий	Выполнен вход в систему	Только системные
13 декабря 2017 в 13:28	Терехов Григорий	Выполнен вход в систему	Кроме системных
11 декабря 2017 в 15:06	Терехов Григорий	Изменены настройки системы	Анна Васильевна
11 декабря 2017 в 14:45	Терехов Григорий	Выполнен вход в систему	Терехов Григорий
11 декабря 2017 в 14:44	Терехов Григорий	Изменены настройки системы	Эльмира Радиковна
11 декабря 2017 в 14:40	Терехов Григорий	Группа отредактирована	Очистить всю историю
11 декабря 2017 в 14:40	Терехов Григорий	Группа отредактирована	
11 декабря 2017 в 14:40	Терехов Григорий	Группа отредактирована	
11 декабря 2017 в 14:38	Терехов Григорий	Группа отредактирована	
11 декабря 2017 в 14:38	Терехов Григорий	Группа отредактирована	
11 декабря 2017 в 14:36	Терехов Григорий	Изменены настройки системы	
11 декабря 2017 в 14:22	Терехов Григорий	Выполнен вход в систему	
11 декабря 2017 в 14:19	Терехов Григорий	Изменения пользователя сохранены	
11 декабря 2017 в 14:19	Терехов Григорий	Создан новый пользователь	

Рис. 38. Просмотр логов пользователей

Специальная страница доступна только для разработчика и руководства организации. Отображается полную таблицу логов (в т.ч. системные) действий пользователей. Имеется возможность сортировать логи по их типу и пользователю. Само логирование действий осуществляется в автоматическом режиме информационной системой.

«Яндекс.Маркет» - не является страницей для пользователя, однако является важной частью веб-приложения. Позволяет формировать перечень товаров для их отправки в Яндекс.Маркет (рекламная площадка компании Яндекс), включения и настройка данной интеграции происходит в разделе «Администрирование». В каждой группе товаров есть специальная опция «Отображать на Яндекс.Маркет», служащая для определения, какие группы товаров будут продаваться на рекламной площадке.

Таким образом пользователь веб-приложения получает широкий функционал для работы с базой данных торговой организации, а также инструменты по автоматизации и обработке, помогающие вести торговую

деятельность на более качественном уровне с меньшими затратами человеческих ресурсов.

**Минимальные требования к аппаратному и программному обеспечению:**

- Для сервера с 20 подключёнными пользователями:
  - Процессор - Intel Core i3-2xxx, либо аналогичный
  - Оперативная память - 1 гб
  - Жёсткий диск - 1 гб
  - Интернет соединение - 8 мбит
  - Выделенный IP-адрес
  - PHP версии  $\geq 5.6$
  - MySQL server версии  $\geq 5.7$
  - MySQL client версии  $\geq 5.7$
  - Apache  $\geq 2.4$
- Для пользователя:
  - Оперативная память — 1 гб
  - Платформа — 32 или 64 разрядная
  - Операционная система — с графическим интерфейсом
  - Интернет соединение — 2 мбит
  - Браузер — поддерживаемая разработчиком версия, поддержка JavaScript

Вопрос лицензирования важная составляющая проектирования и разработки информационных систем и программных продуктов, которая часто опускается при создании информационных систем частными разработчиками. В данной области главным образом фигурирует понятие автор, который может обладать исключительным правом (имущественное право на использование и распоряжение произведением) на разработанную информационную систему, так и неимущественным правом (право на имя, право на неприкосновенность произведения и другое). Автором является физическое лицо, творческая деятельность которого привела к созданию

произведения, также возможно соавторство. Правообладателем же, может быть сам автор, либо организация, получившая исключительные права от автора по договору отчуждения, либо, если произведение создавалось по служебному заданию. Также исключительное право можно получить по наследству. Однако, неимущественные права остаются за автором навсегда [27].

В качестве правового инструмента, который определяет и распространяет программное обеспечение, выступают лицензии на программное обеспечение. В рамках закона, отсутствие лицензии на программное обеспечение запрещает таковое использовать иным лицам, кроме автора. Лицензии на ПО делятся на свободные и несвободные (проприетарные). Поэтому проектируя информационную систему, необходимо заранее озаботиться вопросом лицензирования будущего произведения. Для лицензирования данной информационной системы было решено использовать *open-source* лицензию свободного и открытого ПО.

Одной из популярных лицензий открытого и свободного ПО является GNU General Public License (GPL). GNU General Public License или открытое лицензионное соглашение GNU — это лицензия, первый вариант которой датируется 1 февраля 1989 года. GNU расшифровывается как «GNU's not UNIX» — это рекурсивный акроним, придуманный Ричардом Столлманом, известным идеологом открытого и свободного программного обеспечения. Такое название было придумано для операционной системы, которую в 80-х годах разрабатывал Столлман. На сегодняшний день существует четыре варианта лицензии [26]. (<http://www.gnu.org/licenses/gpl.html>)

Для применения лицензии GNU GPL v3 к информационной системе достаточно добавить 2 файла к проекту: COPYING (текст лицензии) и README (указание авторства, название произведения и информации о применении лицензии), а также соблюсти главное правило свободного ПО — это доступность информационной системы в открытом доступе [22].

Таким образом в целом разработанная система отвечает критериям открытых систем, код веб-приложения (основного компонента системы) написан с использованием современных стандартов и методологий. Информационная система прошла тестирование, отладку и лицензирование в качестве свободно распространяемого программного обеспечения.

## Заключение

Анализ потребностей коммерческих организаций помог выявить недостатки существующих предложений на рынке программного обеспечения. Было решено создать информационную систему решающую потребность торговой организации в организации хранения, обработки и наполнения информации с целью улучшить эффективность бизнес-процессов. При проектировании и создании были изучены и применены современные стандарты, это позволило соответствовать конечному продукту актуальным требованиям и иметь возможность расширяться и дополняться новыми функциями в будущем. Выбор реализации и компонентов информационной системы проводился с расчётом создать систему, в которой решены полностью или частично недостатки существующих предложений рынка ПО.

Созданная информационная система учитывает бюджетные возможности организаций с низким финансированием информатизации - это выражается в низких аппаратных требованиях и использовании только свободно-распространяемого программного обеспечения. При необходимости аппаратную составляющую можно модифицировать с целью увеличения ресурсов для большего числа пользователей, при этом, внесение изменений в программные компоненты ИС не требуется. Используемое ПО кроссплатформенное, поэтому может устанавливаться на серверном оборудовании с различными операционными системами. Такие характеристики значительно упрощают внедрение информационной системы в организацию. Аналогичные преимущества имеются и на стороне пользователя. Реализация интерфейса посредством веб-приложения позволяет работать с системой на любом персональном компьютере с графическим интерфейсом, интернет-обозревателем и активным интернет-соединением. Такой подход избавляет конечного пользователя от

необходимости устанавливать дополнительное программное обеспечение и даёт доступ к информационной системе повсеместно.

Веб-приложение, отвечающее за интерфейс и алгоритмы обработки данных, является гибким и легко настраиваемым компонентом системы. Благодаря открытому исходному коду, любой разработчик при необходимости может модифицировать веб-приложение под специфические потребности организации. Реализация с учётом современных стандартов и методологий позволит осуществить такие модификации, изменив только необходимый минимум компонентов, а также быстро разобраться в алгоритмах и процессах приложения сторонним разработчикам.

Цели и задачи достигнуты. Созданная информационная система предоставляет возможность торговой организации решить ряд проблем с хранением и наполнением базы данных, автоматизировать обработку данных, сократить время и ресурсы для анализа и аналитики, лучше координировать действия сотрудников и следить за показателями торговой деятельности в реальном времени. Внедрение такой информационной системы упростит достижение целей торговой организации, откроет новые возможности и повысит показатели эффективности коммерческой деятельности.

## Список источников и литературы

1. Айзенберг Б., Кварто вон Тивадар Д., Лайза Т.Д. Тестирование и оптимизация веб. - сайтов. Руководство по Google Website Optimizer. - М.: Вильямс, 2017. - 336 с.
2. Арнольд М., Джефф Д., Клинт М. Администрирование Apache. - М.: Лори, 2016. - 418 с.
3. Белов Г.В. Информационные технологии предпринимательства. - М.: АКАДЕМКНИГА, 2015. - 432 с.
4. Бройдо В.Л., Ильина О.П. Вычислительные системы, сети и телекоммуникации. - СПб: Питер, 2015. - 560 с.
5. Веллинг Л., Томсон Л. Разработка веб. - приложений с помощью PHP и MySQL. - М.: Вильямс, 2016. - 848 с.
6. Глушаков с.В., Жакин И.А., Хачиров Т. с. Программирование Web. - страниц. - М.: Фолио, 2016. - 390 с.
7. Дронов, Прохоренок. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web. - мастера. - СПб: ВХВ-Петербург, 2015. - 768 с.
8. Емельянова Н., Партыка Т., Попов И. Устройство и функционирование информационных систем. – М.: Форум, Инфра, - М, 2016. - 448 с.
9. Колисниченко Д.Н. PHP и MySQL. Разработка Web. - приложений. - СПб: ВХВ-Петербург, 2017. - 640 с.
10. Кошик А. Веб. - аналитика 2.0 на практике. Тонкости и лучшие методики. - М.: Вильямс, 2016. - 528 с.
11. Петин В. API Яндекс, Google и других популярных веб. - сервисов. Готовые решения для вашего сайта. - СПб: ВХВ-Петербург, 2017. - 480 с.
12. Полынская Г. Информационные системы маркетинга. Учебник и практикум.- М.: Юрайт, 2014. - 370 с.
13. Поляков Е.В. PHP на примерах. - М.: Наука и техника, 2017. - 256 с.
14. Райгородский А. Модели Интернета. - М.: Интеллект, 2013. - 64 с.

15. Робачевский А. Интернет изнутри. Экосистема глобальной сети. - М.: Альпина Паблишер, 2017. - 224 с.
16. Робин Н. Создаем динамические веб. - сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. - СПб: Питер, 2016. - 688 с.
17. Скляр Д. Изучаем PHP 7. Руководство по созданию интерактивных веб. - сайтов. М.: Диалектика, 2017. - 464 с.
18. Тихонов А.Ф., Тихонова Л.Н. Visual FoxPro 5.0. - М.: БИНОМ, 2014. - 464 с.
19. Уильям с. Справочник администратора. Internet Information Services (IIS) 7.0. - СПб: Русская Редакция, ВХВ-Петербург, 2014. - 665 с.
20. Ульман Л. PHP и MySQL. Создание интернет. - магазинов. - М.: Вильямс, 2015. - 544 с.
21. Филиппов В. Электронные хранилища информации и WEB. - технологии. - М.: Отдельное издание, 2017. - 80 с.
22. Авторское право, свободные лицензии и GPL с нуля [Электронный ресурс]. - URL: <https://geektimes.ru/post/156615/> (дата обращения: 15.11.2017).
23. БЭМ – технологии Яндекса [Электронный ресурс]. - URL: <https://tech.yandex.ru/bem/> (дата обращения: 29.11.2017).
24. Вебмастерам – технологии Яндекса [Электронный ресурс]. - URL: <https://tech.yandex.ru/hubs/webmasters/> (дата обращения: 23.11.2017).
25. Инструменты разработчика Firefox [Электронный ресурс]. - URL: <https://developer.mozilla.org/ru/docs/Tools> (дата обращения: 29.11.2017)
26. Лицензия для вашего open-source проекта [Электронный ресурс]. - URL: <https://habrahabr.ru/post/243091/> (дата обращения: 19.11.2017).
27. Открытые (свободные) лицензии (open source): история и действие в условиях российского права после 01.10.2014 [Электронный ресурс]. - URL: <http://www.klerk.ru/law/articles/403936/> (дата обращения: 11.11.2017).
28. Перевод стандартов PSR-0, PSR-1, PSR-2, PSR-3, PSR-4 [Электронный ресурс]. - URL: [https://svyatoslav.biz/misc/psr\\_translation/](https://svyatoslav.biz/misc/psr_translation/) (дата обращения: 26.11.2017).



29. Помощь вебмастеру [Электронный ресурс]. - URL: <https://yandex.ru/support/webmaster/index.html> (дата обращения: 23.11.2017).
30. Различные механизмы клиент-серверного взаимодействия в web-приложениях [Электронный ресурс]. - URL: <https://www.ibm.com/developerworks/ru/library/wa-aj-ajaxcomm/> (дата обращения: 19.11.2017).
31. Справочник JavaScript [Электронный ресурс]. - URL: <http://javascript.ru/manual> (дата обращения: 23.11.2017).
32. Справочник по HTML [Электронный ресурс]. - URL: <http://htmlbook.ru/HTML> (дата обращения: 22.11.2017).
33. Справочник jQuery [Электронный ресурс]. - URL: <https://gitcode.ru/manuals/jquery> (дата обращения: 23.11.2017).
34. Стандарты PSR [Электронный ресурс]. - URL: <https://myrusakov.ru/php-psr-standards.html> (дата обращения: 12.11.2017).
35. CSS справочник [Электронный ресурс]. - URL: <http://css.manual.ru/> (дата обращения: 22.11.2017).
36. Google для веб-мастеров – поддержка, обучение, взаимодействие и использование Search Console – Google [Электронный ресурс]. - URL: [https://www.google.ru/intl/ru/webmasters/#?modal\\_active=none](https://www.google.ru/intl/ru/webmasters/#?modal_active=none) (дата обращения: 26.11.2017).
37. HTML 5 [Электронный ресурс]. - URL: <http://htmlbook.ru/html5> (дата обращения: 22.11.2017).
38. PHP Standards Recommendations [Электронный ресурс]. - URL: <http://www.php-fig.org/psr/> (дата обращения: 25.10.2017).
39. PHP: Справочник языка - Manual [Электронный ресурс]. - URL: <http://php.net/manual/ru/langref.php> (дата обращения: 23.11.2017).
40. World Wide Web Consortium (W3C) Россия [Электронный ресурс]. - URL: <http://w3c.org.ru> (дата обращения: 10.11.2017).

**«./index.php» – главный файл веб-приложения.**

```
<?php
```

```
/* Сервер по умолчанию может не отображать ошибки выполнения php,
следующие 2 директивы локально (при выполнении этого скрипта и кода
подключенного этим скриптом) задают обратные настройки. При
необходимости отображения ошибок следует их сделать действующими
(раскомментировать) - это может пригодиться при отладке и тестировании веб-
приложения. */
```

```
ini_set("display_errors",1);
```

```
error_reporting(E_ALL); /* Какого уровня ошибки показывать */
```

```
/* Указываем создать сессию, если она ещё не создана (по ней
идентифицируется браузер клиента) */
```

```
session_start();
```

```
/* Заголовок указывающий в какой кодировке работаем. Указана кодировка
UTF-8/ */
```

```
header('Content-Type: text/html; charset=utf-8');
```

```
/* Задаём часовой пояс, в котором мы работаем */
```

```
date_default_timezone_set('Asia/Yekaterinburg');
```

```
/* Подключается php-код из файла safemysql, он содержит класс safemysql для
работы с базой данных. */
```

```
include_once('components/safemysql/safemysql.class.php');
```

```
/* Подключается php-код из файла functions, он содержит основные PHP
функции системы. */
```

```

include_once('components/functions.php');

/* Подключается php-код из файла config, он содержит основные настройки
системы. */
include_once('components/config.php');

/* Подключается php-код из файла lang, он содержит все языковые константы.
*/
include_once('components/lang.php');

/* Создаю объект $db с экземпляром класса safemysql (при создании класс
подключается к БД, поэтому передаю параметры подключения: пользователь
БД, пароль, название базы, хост) */
$db = new SafeMySQL(['user' => CNF_MYSQL_USER, 'pass' =>
CNF_MYSQL_PASSWORD, 'db' => CNF_MYSQL_DB, 'host' =>
CNF_MYSQL_HOST]);

/* Определяется главная переменная в которой будут храниться все
служебные данные */

/* Выгружаются данные о торговой базе из СУБД */
$sys_config = $db->getRow("SELECT * FROM `config`");

/* Подключается php-код из файла authorization, он производит авторизацию
пользователя */
include_once('components/authorization.php');

/* Записываем get-параметры ЧПУ */
$sys_url = [0 => $_GET['path1'], 1 => $_GET['path2'], 2 => $_GET['path3']];

```

```

/* Запрашиваем данные о запрашиваемой странице из СУБД */
$sys_page = $_GET['path1'];

/* Если никакая страница не запрошена, то загружаем главную */
if($sys_url[0] == null) {
    $sys_url[0] = 'main';
}

$sys_page = $db->getRow("SELECT * FROM `pages` WHERE `link`=?s AND
`rights` LIKE '%?p,%' AND `del` IS NULL LIMIT
1",$sys_url[0],$sys_user['rights']);

/* Если такая страница не существует, то показываем ошибку 404 */
if($sys_page == false) {
    $sys_page = '404';
    $sys_page = $db->getRow("SELECT * FROM `pages` WHERE `link`=?s
AND `rights` LIKE '%?p,%' AND `del` IS NULL LIMIT
1",$sys_page,$sys_user['rights']);
}

/* Загружаем файл с PHP-кодом данной страницы */
include_once('components/pages/'.$sys_page['file'].'.php');

/* Дополнительные данные для вывода в главный шаблон системы */
include_once('components/template_functions.php');

/* Выгружаем данные страницы в главный шаблон системы */
echo data2template('all',
    [
        'page_content' => $sys_page['content'],

```

```

        'page_title' => $sys_page['title'],
        'page_head' => $sys_page['head'],
        'page_foot' => $sys_page['foot'],
        'system_name' => $sys_config['system_name'],
        'system_version' => $sys_config['system_version'],
        'system_version_date' =>
time2date($sys_config['system_version_date'],8),
        'org_name' => $sys_config['org_name'], 'user_name' =>
$sys_user['name'],
        'user_post' => $sys_user['post'],
        'page_add_taskCount' => $sys_page['add_taskCount'],
        'page_add_orderCount' => $sys_page['add_orderCount']
    ]
);
?>

```

**«./templates/all.html» - главный HTML шаблон веб-приложения.**

```

<!DOCTYPE html>
<html lang="ru-RU">
    <head>
        <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
        <meta name="viewport" content="width=1200, maximum-
scale=1.0">
        <title><!--[page_title]--></title>
        <link rel="apple-touch-icon" sizes="180x180"
href="/assets/favicon/apple-touch-icon.png">
        <link rel="icon" type="image/png" sizes="32x32"
href="/assets/favicon/favicon-32x32.png">
        <link rel="icon" type="image/png" sizes="16x16"
href="/assets/favicon/favicon-16x16.png">

```

```

    <link rel="manifest" href="/assets/favicon/manifest.json">
    <link rel="mask-icon" href="/assets/favicon/safari-pinned-tab.svg"
color="#5bbad5">
    <link rel="shortcut icon" href="/assets/favicon/favicon.ico">
    <meta name="apple-mobile-web-app-title" content="Торговая база">
    <meta name="application-name" content="Торговая база">
    <meta
name="msapplication-config"
content="/assets/favicon/browserconfig.xml">
    <meta name="theme-color" content="#ffffff">
    <script src=https://yastatic.net/jquery/1.12.4/jquery.min.js></script>
    <script>window.jQuery || document.write('<script
src=/assets/jquery.min.js></script>')</script>
    <script src=/assets/funcs.js?1></script>
    <link
href="https://fonts.googleapis.com/css?family=Roboto+Condensed:400&subs
et=cyrillic" rel="stylesheet"/>
    <link rel="stylesheet" href="/assets/base.css?13"/>
    <link rel="stylesheet" href="/assets/fontAwesome/font-
awesome.min.css"/>
    <!--[page_head]-->
</head>
<body>
    <div class="width-wrapper">
        <header class="cf">
            <div class="header_top cf">
                <div class="header_menu">
                    <a
href="/config/">Администрирование</a> &nbsp; &nbsp;
                    <a href="/passchange/">Сменить
пароль</a> &nbsp; &nbsp;

```

```

        <a href="/exit/">Выход</a>

    </div>

    <div class="header_title">

        <div          class="header_title_is">Торговая
база</div>

        <div          class="header_title_org"><!--
[org_name]--></div>

    </div>

    <div class="header_user"><!--[user_name]--> (<!--
[user_post]-->)</div>

    <div class="header_info">

        Продажи:    <!--[page_add_orderCount]-->
&nbsp; &nbsp; &nbsp; Задачи: <!--[page_add_taskCount]-->

    </div>

</div>

<ul class="header_pages">

    <li><a
href="/"><div>Главная</div></a></li><li><a
href="/orders/"><div>Заказы</div></a></li><li><a
href="/clients/"><div>Клиенты</div></a></li><li><a
href="/providers/"><div>Поставщики</div></a></li><li><a
href="/goods/"><div>Товары</div></a></li><li><a
href="/tasks/"><div>Задачи</div></a></li><li><a
href="/reports/"><div>Статистика</div></a></li></li>

</ul>

</header>

<h1><!--[page_title]--></h1>

<div class="content cf"><!--[page_content]--></div>

```

```

        <footer class="cf"><a href="/about/">Система управления
торговой базой v.<!--[system_version]--> от <!--[system_version_date]--
></a></footer>

    </div>

    <div id="modalNotice"></div>

    <link
href="https://fonts.googleapis.com/css?family=Open+Sans:400,600,700&sub
set=cyrillic" rel="stylesheet"/>

    <script defer async src=/assets/base.js?5></script>
    <script defer async src=/assets/remPolyfill.js></script>
    <!--[page_foot]-->
    <!--[if lt IE 9]><script src=/assets/html5shiv.min.js></script><script
src=/assets/css3-mediaqueries.js></script><![endif]-->

    </body>
</html>

```

**«./components/functions.php» - файл содержащий пользовательские PHP-функции.**

```

<?php
/* ##### Вставка данных в шаблон */

function data2template($template, $patterns = []) {
    /* Считываем из запрошенного шаблона html-код */
    $template = file_get_contents('templates/'.$template.'.html');
    /* Если получен массив данных для вставки в шаблон, то запускаем
цикл */
    if(is_array($patterns) and $patterns != []) {
        foreach($patterns as $key => $value) {

```



```
/* Вставляем в шаблон данные по ключевым словам,  
заранее прописанными в шаблоне */
```

```
$template = str_replace('<!--['.$key.']-->',$value,$template);
```

```
}
```

```
}
```

```
/* trim - очищает начало и конец переменной от лишних пробелов */
```

```
return trim($template);
```

```
}
```

```
/* ##### Преобразование  
unix_timestamp в понятный человеку формат */
```

```
function time2date($timestamp, $variant) {
```

```
    global $isys_lang;
```

```
    $month = & $isys_lang['time2date']['month'];
```

```
    $month2 = & $isys_lang['time2date']['month2'];
```

```
    $week = & $isys_lang['time2date']['week'];
```

```
    switch($variant) {
```

```
        //1 июля в 12:55
```

```
        case 1:
```

```
            $d=date("j",$timestamp);    $m=$month[date('n',$timestamp)];
```

```
$time=date("H:i",$timestamp); return $d.' '.$m.' в '.$time;
```

```
        break;
```

```
        //1 июля 2014
```

```
        case 2:
```

```
            $d=date('j',$timestamp);    $m=$month[date('n',$timestamp)];
```

```
$y=date('Y',$timestamp);return $d.' '.$m.' '.$y;
```

```
        break;
```

```

//1 июля 2014 в 12:55
case 3:
    $d=date('j',$timestamp);    $m=$month[date('n',$timestamp)];
$y=date('Y',$timestamp); $time=date('H:i',$timestamp); return $d.' '.$m.' '.$y.' в
'.$time;

    break;

//1 июля
case 4:
    $d=date('j',$timestamp);    $m=$month[date('n',$timestamp)];
return $d.' '.$m;

    break;

//Пятница, 1 июля
case 5:
    $d=date('j',$timestamp);    $m=$month[date('n',$timestamp)];
$w=$week[date('N',$timestamp)]; return $w.', '.$d.' '.$m;

    break;

//Июль 2014
case 6:
    $m=$month2[date('n',$timestamp)];    $y=date('Y',$timestamp);
return $m.' '.$y;

    break;

//23:55
case 7:
    return date('H:i',$timestamp);

    break;

//01.02.2017
case 8:
    return date('d.m.Y',$timestamp);

    break;

//18 ноября 22:07

```

```

        case 9:
            $d=date('j',$timestamp);    $m=$month[date('n',$timestamp)];
$time=date('H:i',$timestamp); return $d.'&nbsp;'.$m.'&nbsp;'.$time;
            break;
            //18 ноября 22:07, Пятница
        case 10:
            $d=date('j',$timestamp);    $m=$month[date('n',$timestamp)];
$time=date('H:i',$timestamp);    return    $d.'&nbsp;'.$m.'&nbsp;'.$time.',
'.$week[date('N',$timestamp)];
            break;
            //1 июля в 12:55
        case 11:
            $d=date('j',$timestamp);    $m=$month[date('n',$timestamp)];
$time=date('H:i',$timestamp); return $d.' '.$m.' в '.$time;
            break;

        default: return 'неизвестный формат времени';
    }
}

/* ##### Логирование действий
пользователей */

function user_log($action, $type, $info = null) {
    global $db;
    global $isys_user;
    $db->query("INSERT INTO `logs` SET `time`=unix_timestamp(now()),
`user`=?"i,`action`=?"i,`type`=?"i,`info`=?"s",$isys_user['id'], $action, $type, $info);
}

```

```
/* ##### Определяем, является ли
строка email-адресом */
```

```
function is_email($str) {
    return preg_match("/[a-z\d]\w*@[a-z\d]\w*\.[a-z]{2,4}/i",$str);
}
```

```
/* ##### Суммы прописью */
```

```
function num2str($num) {
    $nul='ноль';
    $ten=array(
        array('один','два','три','четыре','пять','шесть','семь',
'восемь','девять'),
        array('одна','две','три','четыре','пять','шесть','семь',
'восемь','девять'),
    );
    $a20=array('десять','одиннадцать','двенадцать','тринадцать','четырнадцат
ь','пятнадцать','шестнадцать','семнадцать','восемнадцать','девятнадцать');
    $tens=array(2=>'двадцать','тридцать','сорок','пятьдесят','шестьдесят','сем
ьдесят','восемьдесят','девяносто');
    $hundred=array('сто','двести','триста','четыреста','пятьсот','шестьсот',
'семьсот','восемьсот','девятьсот');
    $unit=array( // Units
        array('копейка','копейки','копеек', 1),
        array('рубль','рубля','рублей', 0),
        array('тысяча','тысячи','тысяч', 1),
        array('миллион','миллиона','миллионов', 0),
        array('миллиард','миллиарда','миллиардов', 0),
    );
    //
```

```

list($rub,$kop) = explode('.',sprintf("%015.2f", floatval($num)));
$out = array();
if (intval($rub)>0) {
    foreach(str_split($rub,3) as $uk=>$v) { // by 3 symbols
        if (!intval($v)) continue;
        $uk = sizeof($unit)-$uk-1; // unit key
        $gender = $unit[$uk][3];
        list($i1,$i2,$i3) = array_map('intval',str_split($v,1));
        // mega-logic
        $out[] = $hundred[$i1]; # 1xx-9xx
        if ($i2>1) $out[] = $tens[$i2].' '.$ten[$gender][$i3]; # 20-99
        else $out[] = $i2>0 ? $a20[$i3] : $ten[$gender][$i3]; # 10-19 | 1-
9
        // units without rub & kop
        if ($uk>1) $out[] =
morph($v,$unit[$uk][0],$unit[$uk][1],$unit[$uk][2]);
    } //foreach
}
else $out[] = $nul;
$out[] = morph(intval($rub), $unit[1][0],$unit[1][1],$unit[1][2]); // rub
$out[] = $kop.' '.morph($kop,$unit[0][0],$unit[0][1],$unit[0][2]); // kop
return trim(preg_replace('/ {2,}/', ' ', join(' ', $out)));
}
/* ##### Склонение словоформы */
function morph($n, $f1, $f2, $f5) {
    $n = abs(intval($n)) % 100;
    if ($n>10 && $n<20) return $f5;
    $n = $n % 10;
    if ($n>1 && $n<5) return $f2;
    if ($n==1) return $f1;

```

```

        return $f5;
    }
    /* ##### Конвертер строки в ссылку */
    function str2url($string) {
        $string = mb_strtolower($string);
        $pattern = array(
            'a' => 'a', 'б' => 'b', 'в' => 'v',
            'г' => 'g', 'д' => 'd', 'е' => 'e',
            'ё' => 'e', 'ж' => 'zh', 'з' => 'z',
            'и' => 'i', 'й' => 'y', 'к' => 'k',
            'л' => 'l', 'м' => 'm', 'н' => 'n',
            'о' => 'o', 'п' => 'p', 'р' => 'r',
            'с' => 's', 'т' => 't', 'у' => 'u',
            'ф' => 'f', 'х' => 'h', 'ц' => 'c',
            'ч' => 'ch', 'ш' => 'sh', 'щ' => 'sch',
            'ь' => null, 'ы' => 'y', 'ъ' => null,
            'э' => 'e', 'ю' => 'yu', 'я' => 'ya',
            '(' => null, ')' => null, '[' => null,
            ']' => null, '{' => null, '}' => null,
            '/' => null, '\\' => null, '|' => null
        );
        $string = strtr($string, $pattern);
        $string = preg_replace('~[^\a-z0-9_]+\~u', '-', $string);
        $string = preg_replace('/(?:[^\a-z0-9_])(?<=-)+/i', '', $string);

        $string = trim($string, "-");
        return $string;
    }
    ?>

```

«./assets/base.css» - файл содержащий CSS правила.

```

@charset "utf-8";

.cf:before,.cf:after {content:" ";display:table;}.cf:after {clear:both;}.cf {*zoom:1;}

html {font:normal normal 16px 'Open Sans',sans-serif;}
body {background:#f0f0f0;}

.clear{clear:both;}

a {transition:all linear 0.3s;color:#840000;}
a:hover {color:#E42106 !important;}

blockquote {font:italic normal 0.9rem 'Verdana',sans-serif;color:#333;}

.bold {font-weight:600;color:#222;}

.underline {text-decoration:underline;}

.hidden {display:none !important;}

input.placeholder,textarea.placeholder,select.placeholder {color:#808080
!important;}

::-webkit-input-placeholder{color:#808080;}::-moz-placeholder{color:#808080;}:-
moz-placeholder{color:#808080;}:~ms-input-placeholder{color:#808080;}
input[placeholder]{text-overflow:ellipsis;}input::-moz-placeholder{text-
overflow:ellipsis;}input:-moz-placeholder{text-overflow:ellipsis;}input:-ms-input-
placeholder {text-overflow:ellipsis;}

:~focus::-webkit-input-placeholder{color:#969696}:~focus::-moz-
placeholder{color:#969696}:~focus:-moz-placeholder{color:#969696}:~focus:-ms-
input-placeholder{color:#969696;}

ul.defUI-bull li {list-style:disc outside;}
ul.defUI-num li {list-style:decimal outside;}
ul.defUI-square li {list-style:square outside;}
ul.defUI-circle li {list-style:circle outside;}
ul.defUI-bull,ul.defUI-num,ul.defUI-square,ul.defUI-circle {margin-bottom:10px;}

```

```
ul.defUI-bull li,ul.defUI-num li,ul.defUI-square li,ul.defUI-circle li {margin-left:18px;}
```

```
.defForm {width:500px;box-sizing:border-box;border:solid 1px #e6e6e6;padding:10px 20px 0;margin-bottom:10px;}
```

```
.defFormSave {float:right;margin: 10px 0px -15px 0;background:#49CC3D;border:none;padding: 5px 20px;font-weight: 600;font-size: 0.95rem;transition: 0.2s;color: #fff;}
```

```
.defFormSave:hover {cursor:pointer;background: #39BA2D;}
```

```
.defInput {width:436px;border:solid 1px #f7f7f7;background-color:#f7f7f7;font-size: 0.95rem;padding: 3px 10px;margin-top:3px;}
```

```
.defInput:focus {border-color:#000;background-color:#fff;}
```

```
.defSelect {width:458px;border:solid 1px #f7f7f7;background-color:#f7f7f7;font-size: 0.95rem;padding: 3px 10px;margin-top:3px;}
```

```
.defSelect:focus {border-color:#000;background-color:#fff;}
```

```
.defTextarea {width:458px;border:solid 1px #000;font-size: 0.95rem;padding: 3px 10px;margin-top:3px;}
```

```
.defRadio {font-size: 0.95rem;padding: 3px 10px;margin-top:3px;margin-right:2px;}
```

```
.defInputWrap {margin-bottom:15px;}
```

```
.defInputLabel {font-size:0.9rem;font-weight:600;margin-left:3px;}
```

```
.defButton {margin: 20px 15px 15px 0;background: #fff;border: solid 2px #414aff;padding: 4px 20px;font-weight: 600;font-size:0.95rem;transition:0.2s;}
```

```
.defButton:hover {cursor:pointer;background:#414aff;color:#fff;}
```

```
.defTable {width:100%;box-sizing:border-box;}
```

```
.defTable caption {font-size:1.05rem;font-weight:600;margin:15px 0 10px 0;color:#111;text-align:left;}
```

```
.defTable td {padding:2px 10px;border:solid 1px #dadada;font-size:0.95rem;}
```



```

.defTable thead td {font-weight:600;border-bottom:solid 1px #b1b1b1;}
.defTable tbody tr:nth-child(2n) {background:#f8f8f8;}
.defTable tbody tr:hover {background:#ffe99b;}

.width-wrapper {width:1200px;margin:0 auto;}

header {height:84px;background-color:#fff;box-shadow:0 0 1px 1px
#e6e6e6;margin-top:15px;}
.header_top {height:44px;border-bottom:solid 1px #dad8d8;position:relative;}
.header_title {float:left;padding:0 15px 0;font-weight: 600;margin-top:4px;}
.header_title_is {font-size:0.85rem;}
.header_title_org {font-size:0.8rem;}
.header_user {float:left;margin: 12px 0 0 40px;font-size:0.8rem;}
.header_info {margin:11px 40px 0 40px;font-size:0.9rem;float:left;}
.header_menu {background:#fff;margin:9px 20px 0
0;right:0;position:absolute;transition:all ease-out 120ms;}
.header_menu a {font-size:0.8rem;color:#7A0F04 !important;}

.header_pages {font-size:0.95rem;}
.header_pages a {color: #000 !important;}
.header_pages li.current a {color: #C51B00 !important;}
.header_pages li {display:inline-block;padding:0;margin:0;width:171px;}
.header_pages li div {text-align:center;width:170px;box-sizing:border-
box;padding:8px 0 9px;border-right:solid 1px #e6e4e4;}
.header_pages li div:hover {background:#f7f7f7;}
.header_pages li:last-child div {border-right:none;width:174px;}

h1 {font-size:1.2rem;font-weight:normal;padding:0;margin:13px 0;text-
align:center;text-shadow: 1px 1px #f6f6f6;}

```

```

.content {background:#fff;border-radius:2px;padding:20px;min-height:500px;box-
shadow:0px 0px 1px 1px #ededed;}
.content .wrap-h1 {font-size:1.25rem;font-weight:normal;border-bottom:1px solid
#CACACA;color:#333;padding:0 0px 5px;margin:10px 0 0 0;}
.content h2 {font:normal normal 1.25rem 'Roboto Condensed';margin:15px 0 10px
0;color:#111;}
.content p {margin-bottom:9px;font-size:0.95rem;}
.content img {max-width:100%;}

.pgChpass-result {font-size:1rem;margin-top:20px;color:red;font-weight:600;}

.pgGoods_aside {float:right;width:295px;padding-top:10px;font-size: 0.95rem;}
.pgGoods_asideDelimiter {margin:15px 0;height: 1px;background: #f2f2f2;}
.pgGoods_asideMenu li {margin-bottom:10px;}
.pgGoods_asideMenu li a {background: #f3f3f3;padding: 1px 8px 2px;border-
radius: 2px;}
.pgGoods_asideNewGoods li {margin-bottom:3px;font-size:0.85rem;}

.pgGoods_main {float:left;width:800px;min-height:400px;padding-
right:50px;border-right:solid 1px #dedede;}
.pgGoods_goodsTable {display:none;}
.pgGoods_groupsTable a {color:#000 !important;}
.pgGoods_groupsTable img, .pgGoods_goodsTable img {vertical-
align:middle;margin:0 10px;}
.pgGoods_groupsTable tr td:nth-child(2), .pgGoods_goodsTable tr td:nth-child(5)
{width:80px;}
.pgGoods_goodsTable tr td:nth-child(3) {word-break:keep-all;}
.pgGoods_groupsTableNum {float: right;font-size: 0.9rem;color:666;}

.pgGoodsEdit_form {margin:0 auto 10px;}

```

```

.pgGroupEdit_form {margin:0 auto 10px;}

.pgBrands_table img, .pgGoods_goodsTable img {vertical-align:middle;margin:0
10px;}
.pgBrands_table tr td:nth-child(2) {width:80px;}

.pgBrand_form {float:right;width:500px;}
.pgBrand_info {float:left;width:620px;display:none;}
.pgBrand_delete {border-color:#ff4141;}
.pgBrand_delete:hover {background-color:#ff4141;}

.pgProviders_table img {vertical-align:middle;margin:0 10px;}
.pgProviders_table tr td:nth-child(6) {width:80px;}

.pgProvider_form {float:right;width:500px;}
.pgProvider_info {float:left;display:none;width:620px;}
.pgProvider_delete {border-color:#ff4141;}
.pgProvider_delete:hover {background-color:#ff4141;}

.pgProvider_add {float:right;margin-top:20px;color:#000;}

.pgClients_tableAll img {vertical-align:middle;margin:0 10px;}
.pgClients_tableAll tr td:nth-child(6) {width:115px;}

.pgClients_favorites {width:100%;}
.pgClients_favorites td {padding:3px 0;}

.pgClient_right {float:right;width:500px;}
.pgClient_left {float:left;width:620px;}
.pgClient_delete {border-color:#ff4141;}

```

```

.pgClient_delete:hover {background-color:#ff4141;}

.pgClients_aside {float:right;width:295px;padding-top:10px;font-size: 0.95rem;}
.pgClients_asideDelimiter {margin:15px 0;height: 1px;background: #f2f2f2;}
.pgClients_asideList {}
.pgClients_asideList li {}
.pgClients_asideList li a {background: #f3f3f3;padding: 1px 8px 2px;border-
radius: 2px;}

.pgClients_main {float:left;width:800px;min-height:400px;padding-
right:50px;border-right:solid 1px #dedede;}
.pgClients_tableActive img {vertical-align:middle;margin:0 10px;}
.pgClients_tableActive tr td:nth-child(4) {width:115px;}

.pgClients_tableNew {width:100%;}
.pgClients_tableNew tr td:first-child {width:130px;}

.pgOrders_tableAll img {vertical-align:middle;margin:0 10px;}
.pgOrders_tableAll tr td:nth-child(1) {text-align:center;}
.pgOrders_tableAll tr td:nth-child(7) {width:80px;}

.pgOrders_aside {float:right;width:295px;padding-top:10px;font-size: 0.95rem;}
.pgOrders_asideDelimiter {margin:15px 0;height: 1px;background: #f2f2f2;}
.pgOrders_asideList {}
.pgOrders_asideList li {}
.pgOrders_asideList li a {background: #f3f3f3;padding: 1px 8px 2px;border-
radius: 2px;}

.pgOrders_main {float:left;width:800px;min-height:400px;padding-
right:50px;border-right:solid 1px #dedede;}

```

```

.pgOrders_tableActive td {font-size:0.9rem;}
.pgOrders_tableActive img {vertical-align:middle;margin:0 10px;}
.pgOrders_tableActive tr td:nth-child(1) {text-align:center;}
.pgOrders_tableActive tr td:nth-child(7) {width:80px;}

.pgOrders_tableStat {width:100%;}
.pgOrders_tableStat tr td:first-child {width:130px;}
.pgOrders_tableStat td {border-bottom: solid 1px #ececec;padding: 2px 0px;}
.pgOrders_tableStat tr:last-child td {border:none;}

.pgOrder_right {float:right;width:350px;}
.pgOrder_addTime {float:right;}
.pgOrder_id {float:left;}
.pgOrder_delimiter {margin:8px 0;height: 1px;background: #f2f2f2;}
.pgOrder_marksForm {border:none;width:100%;padding:0;}
.pgOrder_marks {min-height:130px;background:#f9f9f9;margin-top:5px;}
.pgOrder_marks:focus {background:#fff;}

.pgOrder_left {float:left;width:790px;}

.pgOrder_table {margin: 0 0 40px 0;font-size:0.85rem;}
.pgOrder_table table {width:100%;}
.pgOrder_table td {padding: 8px 15px;border: solid 1px #dadada;}
.pgOrder_table td:first-child {width:150px;}
.pgOrder_table input[type="text"] {width:95%;padding: 2px 10px;margin-top: 3px;}
.pgOrder_table select {padding: 3px 10px;margin-top: 3px;}
.pgOrder_table input[type="submit"] {margin-top:-4px;margin-right:15px;}

.pgOrder_delete {border-color:#ff4141;}

```

```

.pgOrder_delete:hover {background-color:#ff4141;}

.pgLogs_aside {float:right;width:295px;padding-top:10px;font-size: 0.95rem;}
.pgLogs_asideDelimiter {margin:5px 0 10px;height: 1px;background: #f2f2f2;}
.pgLogs_asideMenu li {margin-bottom:10px;}
.pgLogs_asideMenu li a {background: #f3f3f3;padding: 1px 8px 2px;border-
radius: 2px;}
.pgLogs_main {float:left;width:800px;min-height:400px;padding-
right:50px;border-right:solid 1px #dedede;}
.pgLogs_table {width:100%;}
.pgLogs_table thead td {font-weight:600;}
.pgLogs_table caption {font-size:1.05rem;font-weight:600;margin:15px 0 10px
0;color:#111;text-align:left;}
.pgLogs_table td {padding:2px 10px;border:solid 1px #dadada;font-size:0.95rem;}
.pgLogs_table tr td:nth-child(7) {width:80px;}
.pgLogs_table tr:nth-child(2n) {background: #f8f8f8;}
.pgLogs_table tbody tr:hover {background:#ffe99b;}

.pgGroupEdit_delete, .pgGoodsEdit_delete {border-color:#ff4141;margin-top:
10px;}
.pgGroupEdit_delete:hover, .pgGoodsEdit_delete:hover {background-
color:#ff4141;}

.pgGoodsEdit_delete, .pgGoodsEdit_delete {border-color:#ff4141;}
.pgGoodsEdit_delete:hover, .pgGoodsEdit_delete:hover {background-
color:#ff4141;}

.pgMain_aside {float:right;width:295px;font-size: 0.95rem;}
.pgMain_main {float:left;width:800px;min-height:400px;padding-
right:50px;border-right:solid 1px #dedede;}

```

```

.pgMain_valutesRate {margin-bottom:60px;}
.pgMain_valutesRate_now {text-align:center;margin:20px 0 0;}
.pgMain_valutesRate_chart {}

.pgMain_docs {padding-left:10px;line-height:1.6rem;margin-bottom:40px;}

.pgMain_stat {margin-left:10px;}
.pgMain_stat_block {border:solid 1px #ccc;padding:15px 10px 5px
10px;margin:25px 0 10px 0;font-size:0.9rem;}
.pgMain_stat_block p {font-size:0.9rem;margin-bottom:0;}
.pgMain_stat_title {position: absolute;margin:-25px 0 5px
5px;background:#fff;display:inline-block;padding:0 8px;}

.pgMain_chat {}
.pgMain_chat_msgs {font-size:0.85rem;height: 300px;overflow-y: scroll;border:
solid 1px #ccc;padding: 5px 10px;}
.pgMain_chat_msg {margin-bottom:10px;}
.pgMain_chat_form {}
.pgMain_chat_form input[type="text"] {font-size: 0.9rem;height: 29px;width:
500px;padding-left:10px;}

.pgReports_table {margin-top:50px;}
.pgReports_detailTable {display:none;width:100%;margin-bottom:40px;}
.pgReports_detailForm select {width:110px;}

.pgTasks_table {}
.pgTasks_table tr.active a {font-weight: 600;}

.pgTask_aside {float:right;width:375px;font-size: 0.95rem;}

```

```

.pgTask_aside p {margin-bottom:8px;padding-bottom:8px;border-bottom:solid 1px
#D6D6D6;}

.pgTask_main {float:left;width:730px;min-height:400px;padding-
right:35px;border-right:solid 1px #dedede;}

.pgTask_main h2 {border-bottom:solid 1px #ccc;padding-bottom:3px;margin-
top:30px;}

.pgTask_main h2:first-child {margin-top:0;}

.pgTask_main h2:nth-of-type(3) {border-bottom:none;}

.pgTask_asideList {margin-left:10px;}

.pgTask_asideList li {margin-bottom:8px;}


.pgTask_delete {display:none;margin-left: 100px;}


.pgTask_chat {}

.pgTask_chat_msgs {font-size:0.85rem;height: 300px;overflow-y: scroll;border:
solid 1px #ccc;padding: 5px 10px;}

.pgTask_chat_msg {margin-bottom:10px;}

.pgTask_chat_form {}

.pgTask_chat_form input[type="text"] {font-size: 0.9rem;height: 29px;width:
500px;padding-left:10px;}


.pgTask_aside .defSelect {width:350px;}

.pgTask_aside .defInput {width:330px;}

.pgTask_checkBox {margin:5px 0;}

.pgTask_checkBox {}

.pgTask_save {margin:20px 0;float:left;}

.pgTask_edit {display:none;margin:30px 0;color:#000;}

.pgTask_add {float:right;margin-top:20px;color:#000;}


.pgConfig_setWrap {float:left;vertical-align:top;}

```



```

.pgConfig_setWrap .defForm {margin-bottom:20px;padding-top:0;}
.pgConfig_usersWrap {float:right;width:645px;padding-top:0;vertical-align:top;}
.pgConfig_user {padding-bottom:25px;margin:35px 0;border-bottom:dashed 1px
#ccc;}
.pgConfig_user .defFormSave {font-size: 0.85rem;padding: 4px 10px;margin:10px
250px 0 0;}
.pgConfig_newUserWrap {float:right;width:645px;padding-top:0;vertical-
align:top;margin-top:10px;}
.pgConfig_newUserWrap .defFormSave {font-size: 0.85rem;padding: 4px
10px;margin:10px 250px 20px 0;}
.pgConfig_userTitle {margin-bottom:10px;}
.pgConfig_userTitle img {vertical-align:middle;}
.pgConfig_userInputs {margin-bottom:10px;font-size:0.9rem;}
.pgConfig_userInputs input {width:130px;height:18px;border:solid 1px
#f7f7f7;background-color:#f7f7f7;padding:2px 1px 3px 5px;}
.pgConfig_userInputs select {width:120px;height:26px;border:solid 1px
#f7f7f7;background-color:#f7f7f7;padding: 0px 3px 3px;}
.pgConfig_userInputs input:focus, .pgConfig_userInputs select:focus {border-
color:#000;background-color:#fff;}

.pgConfig_viewLogs {float:right;color:#000;}

footer {margin:10px 15px 30px;font-size:0.8rem;text-align:right;}
footer a {color:#333 !important;}

.ck-editor__editable {
    min-height: 200px;
}

/* modal windows*/

```

```
@keyframes modalOpening {0% {opacity:0.5;box-shadow:0px 0px 4px rgba(19, 75, 171, 0.8);}100% {opacity:1;box-shadow:none;}}
```

```
div#modalNotice {display:none;position:fixed;z-index:900;bottom:15%;left:2%;padding:20px 10px;width:300px;color:#fff;border-radius:2px;font-size:1.1rem;font-weight:400;text-align:center;}
```

```
div#modalNotice.red {background:rgba(120, 15, 15, 0.8);}
```

```
div#modalNotice.blue {background:rgba(15, 79, 120, 0.8);box-shadow:0px 0px 2px 0px rgba(0, 21, 57, 0.4);}
```

**«./components/pages/reports.php» - файл содержащий PHP-код раздела статистики.**

```
<?php
```

```
/* Получаем данные для построения графиков */
```

```
$chart = [];
```

```
$chart_sum = [];
```

```
$chart_count = [];
```

```
$q = $db->query("SELECT `add_time`,`goods_sum` FROM `orders` WHERE `add_time`>?i AND `status`='2' AND `del` IS NULL ORDER BY `id` DESC",strtotime('-100 days'));
```

```
while($ls = $db->fetch($q)) {
```

```
    $ls['add_time'] = strtotime(date('Y-m-d',$ls['add_time']));
```

```
    if(!isset($chart[$ls['add_time']])) {
```

```
        $chart[$ls['add_time']] = ['count' => 0, 'sum' => 0];
```

```
    }
```

```
    $chart[$ls['add_time']]['count']++;
```

```
    $chart[$ls['add_time']]['sum'] += $ls['goods_sum'];
```

```
}
```

```

foreach($chart as $key => $val) {
    $key = time2date($key,2);
    $val['sum'] = round($val['sum']);
    $chart_sum[] = "[".$key.", {$val['sum']}]";
    $chart_count[] = "[".$key.", {$val['count']}]";
}

$chart_sum = implode(',',array_reverse($chart_sum));
$chart_count = implode(',',array_reverse($chart_count));

/* Составление таблиц */

$stable_all = ['count' => 0, 'sum' => 0, 'sum_middle' => 0, 'cancel' => 0, 'user' => [],
'client' => 0];
$stable = [
    '1month' => ['count' => 0, 'sum' => 0, 'sum_middle' => 0, 'cancel' => 0, 'user'
=> [], 'client' => 0, 'date' => null],
    '2month' => ['count' => 0, 'sum' => 0, 'sum_middle' => 0, 'cancel' => 0, 'user'
=> [], 'client' => 0, 'date' => null],
    '3month' => ['count' => 0, 'sum' => 0, 'sum_middle' => 0, 'cancel' => 0, 'user'
=> [], 'client' => 0, 'date' => null]
];

$q = $db->query("SELECT `add_time`,`goods_sum`,`user`,`status` FROM
`orders` WHERE `add_time`>=?i AND (`status`='2' OR `status`='3') AND `del` IS
NULL ORDER BY `id`",strtotime('-2 month',strtotime(date('Y-m-01',time()))));
$stable_month = null;
while($ls = $db->fetch($q)) {
    $add_time_month = strtotime(date('Y-m-01',$ls['add_time']));

```

```

switch($add_time_month) {
    case strtotime('-1 month',strtotime(date('Y-m-01',time()))):
        $table_month = &$tables['2month'];
        break;
    case strtotime('-2 month',strtotime(date('Y-m-01',time()))):
        $table_month = &$tables['3month'];
        break;
    default: $table_month = &$tables['1month'];
}
if(!isset($table_month['date'])) {
    $table_month['date'] = strtotime(date('Y-m-01',$ls['add_time']));
}
if($ls['status'] == 3) { $table_month['cancel']++; continue;}
$table_month['count']++;
$table_month['sum'] += $ls['goods_sum'];
if(isset($table_month['user'][$ls['user']])) {
    $table_month['user'][$ls['user']]++;
}
else { $table_month['user'][$ls['user']] = 1; }
}

foreach($tables as $month_name => &$month) {

    $month['sum_middle']    =    ($month['sum']    ==    0    ?    0    :
round($month['sum']/$month['count']));

    $month['client'] = $db->getOne("SELECT COUNT(`id`) FROM `clients`
WHERE `add_time`>=?i AND `add_time`<=?i AND `del` IS NULL",
$month['date'], strtotime('+1 month',$month['date']));

    $table_all['count'] += $month['count'];

```

```

$stable_all['sum'] += $month['sum'];
$stable_all['sum_middle'] += $month['sum_middle'];
$stable_all['cancel'] += $month['cancel'];
$stable_all['client'] = $db->getOne("SELECT COUNT(`id`) FROM `clients`
WHERE `add_time`>=?i AND `add_time`<=?i AND `del` IS NULL", strtotime('-2
month',strtotime(date('Y-m-01',time()))), time());

foreach($month['user'] as $user_id => $user_count) {
    if(!isset($stable_all['user'][$user_id])) { $stable_all['user'][$user_id] =
$stable_all['count']; }
    else { $stable_all['user'][$user_id] += $user_count; }
}
}
$stable_all['sum_middle'] = round($stable_all['sum']/$stable_all['count']);
$full_table = '
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Продажи | .$stable_all['count'].</td>  .$tables['1month'] ['count'].</td>  .$tables['2month'] ['count'].</td>  .$tables['3month'] ['count'].</td> .(isset($tables['2month'] ['user'][$user_id] $tables['2month'] ['user'][$user_id] : 0).</td>  .(isset($tables['3month'] ['user'][$user_id] $tables['3month'] ['user'][$user_id] : 0).</td> | | | | |

```

```

$full_table .= '
        <tr><td>Оборот</td><td>'.$table_all['sum'].'
p.</td><td>'.$tables['1month']['sum'].'          p.</td><td>'.$tables['2month']['sum'].'
p.</td><td>'.$tables['3month']['sum'].' p.</td></tr>

        <tr><td>Средний                чек</td><td>'.$table_all['sum_middle'].'
p.</td><td>'.$tables['1month']['sum_middle'].'
p.</td><td>'.$tables['2month']['sum_middle'].'
p.</td><td>'.$tables['3month']['sum_middle'].' p.</td></tr>

        <tr><td>Отменённые
заказы</td><td>'.$table_all['cancel'].'</td><td>'.$tables['1month']['cancel'].'</td>
<td>'.$tables['2month']['cancel'].'</td><td>'.$tables['3month']['cancel'].'</td></tr>

        <tr><td>Новых
клиентов</td><td>'.$table_all['client'].'</td><td>'.$tables['1month']['client'].'</td>
<td>'.$tables['2month']['client'].'</td><td>'.$tables['3month']['client'].'</td></tr>
';

```

/\* Добавляем в head страницы скрипт Google Charts \*/

```

$sys_page['head']          =          '<script          type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>';

```

/\* Выбор даты для детального отчёта \*/

```

$select_date = null;
$q = $db->query("SELECT DATE_FORMAT(FROM_UNIXTIME(`add_time`),
'%m-%Y') AS 'month_year' FROM `orders` WHERE `status`='2' AND `del` IS
NULL GROUP BY `month_year` DESC");
while($ls = $db->fetch($q)) {
        $select_date          .=          '<option
value="'.$ls['month_year'].'">'.$ls['month_year'].'</option>';

```

```
}
```

```
/* Если выбрана дата детальной статистики */
```

```
$month_table = null;
```

```
$month_caption = null;
```

```
$month_summ = 0;
```

```
$month_count = 0;
```

```
$month_cancel = 0;
```

```
$month_start = 0;
```

```
$month_end = 0;
```

```
if(!empty($_POST['select_date'])) {
```

```
    $month_start = strtotime('01-'. $_POST['select_date']);
```

```
    $month_end = strtotime(date('t', $month_start). '-'. $_POST['select_date']);
```

```
    $month_caption = time2date($month_start, 6);
```

```
    $sys_page['head'] .= ' <style type="text/css">.pgReports_detailTable  
{display:block;}</style>';
```

```
    $q = $db->query("SELECT  
`id`,`add_time`,`status`,`goods_sum`,`pay`,`delivery`,(SELECT `name` FROM  
`clients` WHERE `id`=`client` LIMIT 1) as 'client',(SELECT `name` FROM  
`users` WHERE `id`=`user` LIMIT 1) as 'user' FROM `orders` WHERE  
`add_time`>=?i AND `add_time`<=?i AND (`status`='2' OR `status`='3') AND  
`del` IS NULL ORDER BY `add_time` ASC, `client`, $month_start,  
$month_end);
```

```
    while($ls = $db->fetch($q)) {
```

```
        if($ls['status'] == 3) { $month_cancel++; continue; }
```

```
        $month_count++;
```

```
        $month_summ += $ls['goods_sum'];
```

```

        $month_table                .=          '<tr><td>'.$ls['id'].'</td><td><a
href="/order_edit/'.$ls['id'].'/">'.time2date($ls['add_time'],9).'

```

/\* Передаём данные в шаблон страницы статистики \*/

```

$sys_page['content'] = data2template('reports', ['chart_count' => $chart_count,
'chart_sum' => $chart_sum,
'table' => $full_table,
'1month' => time2date($tables['1month']['date'],6),
'2month' => time2date($tables['2month']['date'],6),
'3month' => time2date($tables['3month']['date'],6),
'select_date' => $select_date,
'month_table' => $month_table,
'month_caption' => $month_caption,
'month_summ' => $month_summ,
'month_count' => $month_count,
'month_cancel' => $month_cancel
]);
?>

```